ADA073588

[redacted]

AFFDL-TR-79-3011

DIGITAL LOAD CONTROL APPLIED TO FULL-SCALE AIRFRAME FATIGUE TESTS

Nirmal K. Mondol
Richard M. Potter, Major, USAF
Structures Test Branch
Structures and Dynamics Division

May 1979

TECHNICAL REPORT AFFDL-TR-79-3011

Final Report for Period January 1976 - June 1977

Approved for public release; distribution unlimited.

AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (IO) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

NIRMAL K. MONDOL
Electrical Engineer

SANFORD LUSTIG
Chief, Structures Test Branch

FOR THE COMMANDER

RALPH L. KUSTER, JR, Colonel, USAF
Chief, Structures and Dynamics Division

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE/56780/24 August 1979 — 500

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFFDL-TR-79-3011	2. GOVT ACCESSION NO.
	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): DIGITAL LOAD CONTROL APPLIED TO FULL-SCALE AIRFRAME FATIGUE TESTS	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report January 1976 - June 1977
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Nirmal K. Mondol; Richard M. Potter, Maj, USAF	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Structures Test Branch (FBT), Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio 45433	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Project No. 1347, Task No. 03, Work Unit No. 02
11. CONTROLLING OFFICE NAME AND ADDRESS: Structural and Dynamics Division (FB), Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio 45433	12. REPORT DATE: May 1979
	13. NUMBER OF PAGES: 174
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Airframe fatigue tests
Load control
Direct-digital load control
Electro-hydraulic load control
Load (forces)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The modeling, analysis and digital simulation of an analog servo controller and its successful application to a full-scale airframe fatigue test is described. Primary emphasis is on the use of minicomputers for dynamic load control of multiple channels. Hardware and software used to generate functions and control load is described. A brief comparison of digital system performance versus conventional analog controllers is included.

ADA073588

FOREWORD

This is the final report of work accomplished and results achieved at the Test Facility Group, Structures Test Branch, Structures and Dynamics Division, Air Force Flight Dynamics Laboratory, WPAFB, Ohio under in-house Project 1347, Task 03, Work Unit 02, "Digital Load Control."

This report covers work performed between January 1976 and June 1977. Mr. Nirmal K. Mondol was the principal investigator. Maj R. M. Potter, Air Force Institute of Technology, Electrical Engineering Department, provided the design and analysis reported in Sections II and III. Mr. Don Heidorn of Digital Equipment Corporation provided the software under Contract F33615-76-C-0019. Capt Vince Darcy of the Flight Controls Group provided the initial direction towards successful control. Prof. J. J. D'Azzo, Capt J. B. Peterson, and Capt P. E. Miller of the Air Force Institute of Technology, Electrical Engineering Department, assisted with consultation on a formalized approach to system analysis and setup. In-house support was furnished by Mr. Dansen Brown, Flight Analysis Group.

Mr. Adam Grube designed and built the low-level feedback and multiplexer system. Mr. Joseph Pokorski provided all engineering support required for the load-sensing systems. Messrs. Dave Shultz and Roger Orchard provided immediate technical support for the project. Hydraulics support was furnished by Messrs. Keith Fortune and Harold Schaefer. Messrs. Don Brammer, George Holderby and James Specht provided the expertise to get a full-scale airframe fatigue test off to a good start on the first try. Mr. B. C. Boggs was Technical Manager of the Test Facility Group, which had primary responsibility for this task/work unit.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF SYMBOLS

SYMBOL	MEANING
A	Digital Proportional Gain Constant
B	Digital Integral Gain Constant
C	Feedback Signal
DDC	Direct Digital Control
E	Error
K	Combined Hydraulic Jack-Load Gain Constant
K_A	Hydraulic Jack Gain Constant
K_L	Load Gain Constant
K_p	Analog Proportional Gain Constant
K_i	Analog Integral Gain Constant
M_p	Peak Overshoot (Percent)
$M(T_K)$	Controller Output at Time T_K
PI	Proportional-Integral
R	Reference Load
S	Complex Variable $\sigma + j\omega$
T	Sampling Period
T_P	Time to Peak
$X(T_K)$	Integral Portion of Controller Output at Time T_K
X_L	Load Rate
Y	Load Applied to Structure
ω_n	Natural Frequency
ζ	Damping Ratio
UDC	Universal Digital Controller

LIST OF SYMBOLS (Cont'd)

SYMBOL	MEANING
RTP	Real-Time Processor
D/A	Digital-to-Analog Converter
A/D	Analog-to-Digital Converter

SECTION I

INTRODUCTION

This report documents Phase 2 of a continuing effort to simplify test methods and equipment for full-scale airframe fatigue tests specifically in the areas of function generation and electro-hydraulic load control. Function generation is presently accomplished with a wide variety of analog and digital devices. Load control is achieved with analog servo loops (one per load area) which match the program (reference) value with a feedback signal furnished by load cells to regulate the flow of high-pressure hydraulic fluid to the loading jacks.

The effort was initiated to investigate direct-digital load control techniques as a cost effective means of improving and simplifying test methods for closed-loop electro-hydraulic load application and as a possible means of realizing increased hardware reliability.

Technical Memorandum AFFDL-TM-75-89 describes Phase 1 of this work effort to develop a multi-channel load control system suitable for airframe fatigue testing. Phase 1 goals were to (a) investigate programming languages considered standard in the software industry that were easy to use and highly interactive and compatible for load control application, and (b) demonstrate feasibility of a multi-channel direct-digital load control system.

During Phase 1, control algorithms were written successively in assembly language, FOCAL and BASIC. Each of the software packages were tested for performance characteristics on a four-channel test stand. Phase 1 concluded with a feasibility demonstration using an F-111 wing as the test article. The load spectrum was of the constant-amplitude repeat-cycle type. Load application was by means of 13 hydraulic jacks arranged to load in tension only. The feasibility demonstration was accomplished with a single minicomputer performing all necessary tasks, i.e., functioning as a digital servo controller for thirteen channels, generating the load spectrum for thirteen channels, and maintaining an interactive relationship with the operator for on-line changes

to control parameters. Provisions were included for generation of calibration voltages, ramps and sine waves as well as jogging of the jacks to facilitate test setup. Single-user BASIC was used for this demonstration because it offered the on-line interactive capability considered a necessity in a test facility environment.

Two problems were noted during Phase 1, (a) BASIC was too slow to provide the sampling times necessary for a digital servo loop because of its inherent interpretive structure, and (b) the digital controller performance needed improvement.

Phase 2 efforts were directed toward continuing the development of a multi-channel digital load control system capable of performing flight-by-flight aircraft fatigue testing.

The characteristics of a continuous analog Proportional plus Integral (PI) controller were selected for approximation by the digital system. The PI controller was chosen as the model because it is the type most commonly used in the Structures Test Facility.

The digital controller equations and the general range of parameters over which they should be expected to apply are described in Section II of this report. Appendix D contains the derivation of these equations from the differential equation describing an analog (continuous) PI controller.

A methodology for determining "gains" of the digital controller based on the performance of its continuous analog model is described in Section III.

The flow chart of the basic controller software implementation is described in Section IV. Sections V and VI deal with experimental evaluation of the digital controller on a test stand, and Section VII describes its deployment on a full-scale test program after successful completion of the test stand experiments.

All main computer programs in this study were written in FORTRAN IV (Section VII). This provided the digital controller algorithm a period of 10 milliseconds. This was adequate to provide the controller performance described in Sections II, III, IV and V.

Two minicomputers were used in Phase 2 in a master/slave configuration in order to separate the digital control algorithm from load spectrum generation and other miscellaneous tasks. This allowed maximum possible sampling speeds and independence of computing loads imposed by other test requirements.

SECTION II

DIGITAL LOAD CONTROLLER DESIGN

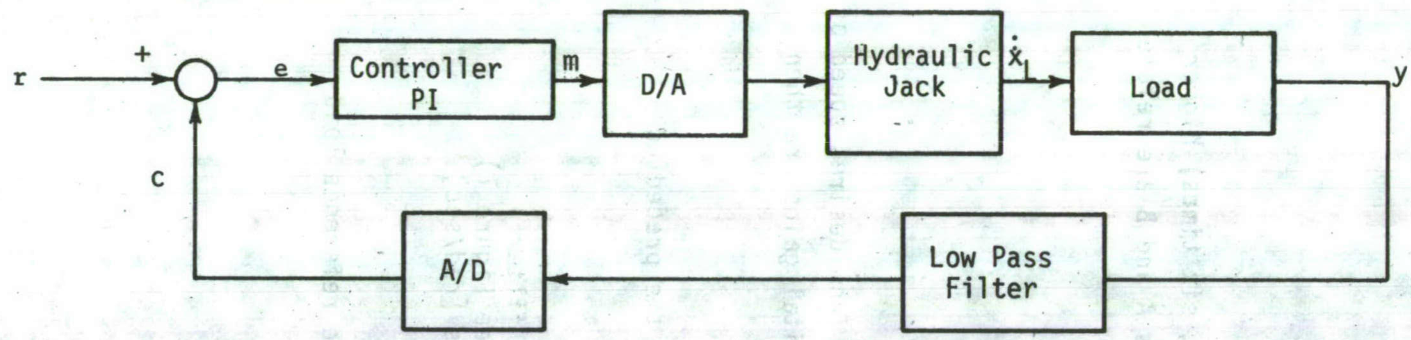
This section describes the derivation of a digital load controller from a continuous proportional-integral model.

MODEL

A functional block diagram of a single channel of the Direct Digital Load Control (DDC) servo loop is presented in Figure 1. Nomenclature is defined on the figure and the relation to variables on the DDC console is shown.

A continuous model that approximates the low-frequency behavior of the system is depicted in Figure 2. The hydraulic jack provides a load rate proportional to drive signal when loop operation is limited to frequencies below its bandwidth (50-100 Hz). The gain relating hydraulic jack drive to load rate is denoted by K_A $[\frac{ft/sec}{counts}.]$ (Counts corresponds to the numerical input into the digital-to-analog converter driving the servovalve; e.g., $\pm$ 4096 counts = $\pm$ 10 Vdc output.) The force applied by the hydraulic jacks is roughly proportional to load displacement provided that loop operation is well below structural resonant frequencies. The associated "spring" constant is K_L $[\frac{counts}{ft}]$. The product $K_A K_L = K$ $[sec^{-1}]$, defined as the open-loop sensitivity, is independent of drive and is required to determine approximate gain settings. This product can be estimated as: SLOPE OF OUTPUT OF LOAD CELL [COUNTS PER SEC] ÷ STEADY HYDRAULIC JACK DRIVE [COUNTS.]

A PI controller is used with proportional and integral gains K_p (dimensionless) and K_I (1/sec). This type of analog controller is currently in facility use on full scale multichannel airframe fatigue tests. For DDC, this controller is implemented with a digital PI controller with integration performed using the trapezoidal rule



Block Diagram	Physically	Display
r	Reference Load	REF
c	Conditioned load measurement	CONTROL
e	Error between measured & commanded load	ERROR
m	Controller output (current to hydraulic jack)	OUTPUT
$\dot{x}_L$	Load Flow (velocity of load)	--
y	Load applied to structure through hydraulic jack	--

Figure 1. Block Diagram of DDC Load Simulator Servo Loop

(Tustin Approximation - See Appendix D). The controller algorithm developed by Darcy[1] is:

$$M(T_K) = X(T_K) + AE(T_K)$$
$$X(T_{K+1}) = X(T_K) + BE(T_K)$$
$$T_K+1 - T_K = T \text{ [sec]}$$

The sample rate, 1/T, should be substantially (e.g., six times) higher than all frequencies of interest. The digital gains A and B are related to the analog gains K_p and K_i through:

$$A = K_p + K_i \; T/2$$

$$B = TK_i$$

The continuous model is used to select analog gains from desired speed of response and peak overshoot. The corresponding digital gains are then obtained from the relations above.

The model depicted in Figure 2 as described in the preceding paragraphs applies when

KpK and $\sqrt{K_i K}$ [sec^{-1}]<<bandwidth of low pass filter in feedback loop (see Figure 1)
bandwidth of actuators [rad/sec]
structural resonance

Low-frequency operation means the frequencies in the reference load satisfy the same restriction as KpK and $\sqrt{K_i K}$.

---

[1]Capt Vince Darcey, Flight Control Division, AFFDL

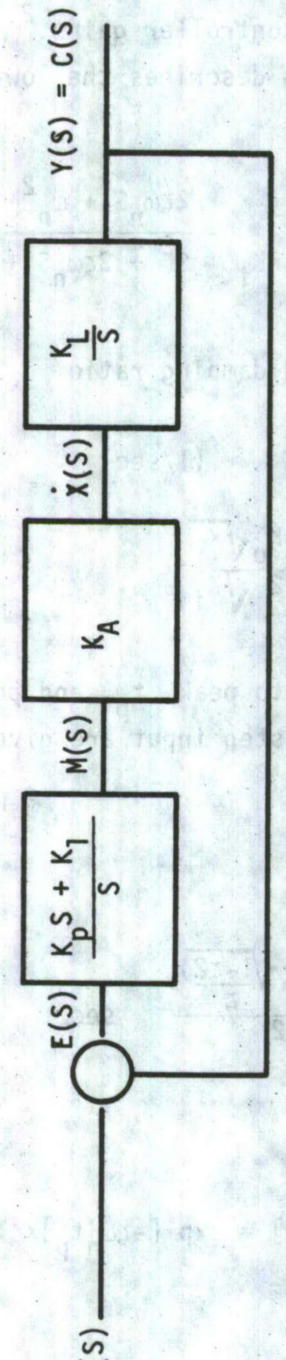


Figure 2. Continuous Model for Low-Frequency Response

SECTION III

DIGITAL CONTROLLER GAIN DETERMINATION

The response of the PI analog model to step inputs is analyzed and used as a basis for selecting controller gains.[2] From Figure 2, the closed-loop transfer function which describes the low-frequence operation is:

$$\frac{Y(s)}{R(s)} = \frac{K(K_p S + K_i)}{S^2 + KK_p\ S + KK_i} = \frac{2\zeta\omega_n S + \omega_n{}^2}{S^2 + 2\zeta\omega_n S + \omega_n{}^2} \tag{1}$$

with undamped natural frequency and damping ratio

$$\omega_n = \sqrt{K_i K} \qquad (1/\text{sec}) \tag{2}$$

$$\zeta = \frac{K_p\sqrt{K}}{2\sqrt{K_i}} \tag{3}$$

Formulas for the theoretical, time to peak, t_p, and percent overshoot, M_p, when the system is forced by a step input are given below:

Case 1 (underdamped, $\zeta<1$

<u>Time to peak</u>

$$t_p = \frac{2\tan^{-1}\left(\frac{\sqrt{1-\zeta 2}}{\zeta}\right)}{\omega_n\ \sqrt{1-\zeta^2}} \qquad \text{sec} \tag{4}$$

<u>Percent Overshoot</u>

$$M_p = \exp\left[-\zeta^2\tan^{-1}\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right) / 1-\zeta^2\right] = \exp\left[-\zeta\omega_n t_p\right] \times 100 = \%\ \text{Overshoot} \tag{5}$$

[2]Reference Linear Control System Analysis and Design, Dazzo and Houpis, McGraw and Hill, 1975.

Case 2 (overdamped, $\zeta>1$

Time to Peak

$$t_p = 1_n\left[\frac{\zeta + \sqrt{\zeta^2-1}}{\zeta - \sqrt{\zeta^2-1}}\right]/\omega_n \sqrt{\zeta^{2-1}} \quad \text{sec} \tag{6}$$

Percent Overshoot

$$Mp = \frac{\zeta + \sqrt{\zeta^2-1}}{\zeta - \sqrt{\zeta^2-1}} -\zeta/ \sqrt{\zeta^2-1} = \exp(-\zeta\omega_n t_p) \times 100 = \% \text{ Overshoot} \tag{7}$$

Graphs relating normalized time-to-peak and percent peak overshoot are presented in Figures 3 and 4. The graphs show that overshoot is modest (less than 20 percent) for $\zeta > .75$. Fatigue testing demands minimum overshoot which was limited to 5 percent for the Advanced Metallic Air Vehicle Wing Carry Through Structure. The acceleration error coefficient is proportional to K_i so steady state tracking error is minimized by increasing K_i. Also for fixed Kp and K the time-to-peak [sec] decreases with increasing K_i provided that $\zeta < 2.2$. Thus, the largest K_i consistent with acceptable damping is recommended.

GAIN SETTINGS

A suggested procedure for determining gain settings is:

1. *Determine K

2. With $K_i = 0$; increase Kp until either instability occurs or Kp K ≈ 3db frequency of hydraulic jack (Hz). Set Kp = $\frac{K_p}{6}$ (critical)

3. Set $K_i = Kp^2 K/4\zeta^2$ (results in a reasonable approximation)

*Either determine the slope of load vs. time in open loop operation or measure percent overshoot for a closed loop operation and use the graph (Figure 3) to determine the corresponding ζ. Then $K = 4K_{I\zeta}^2/Kp^2$.

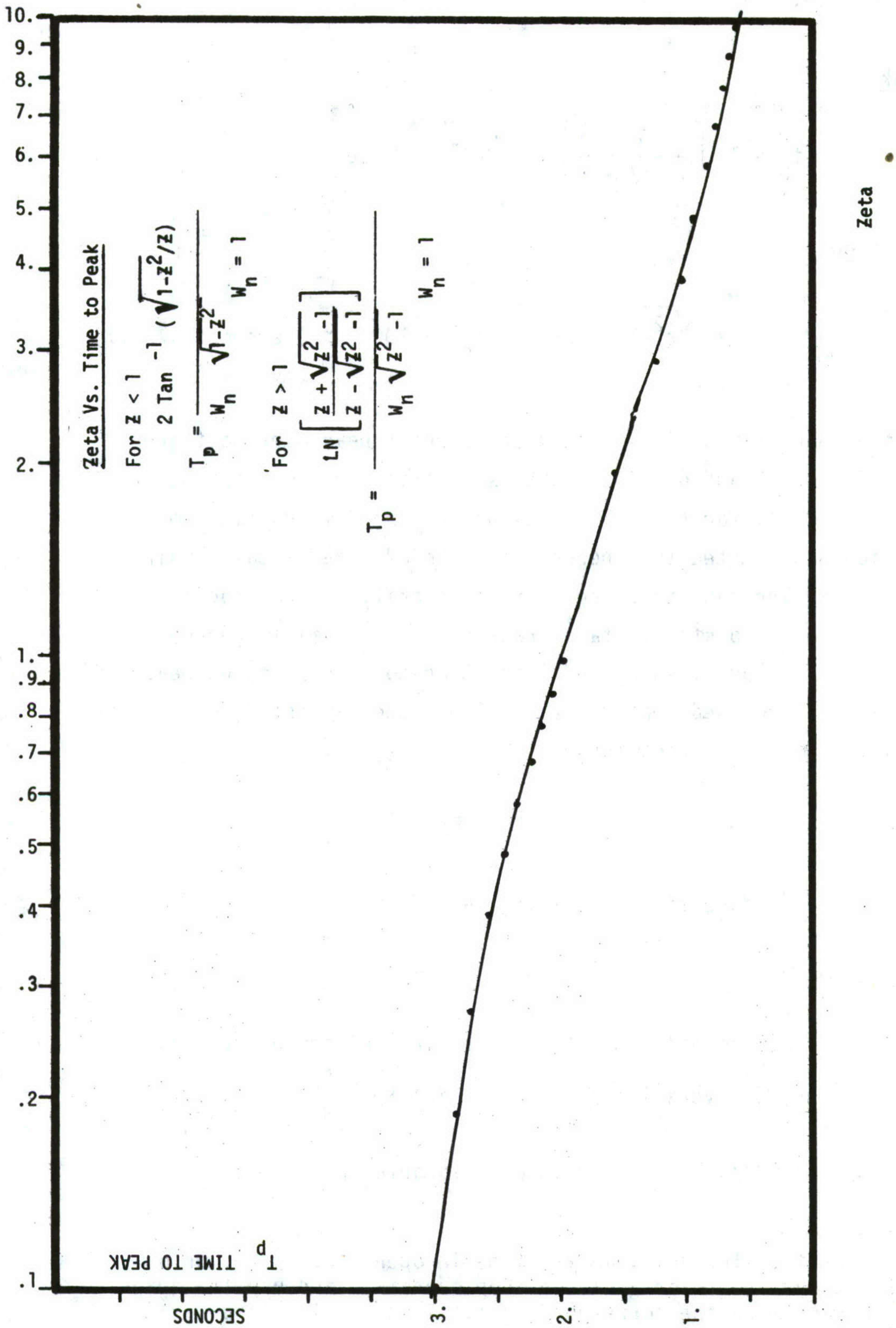


Figure 3. Damping Ratio vs. Time to Peak

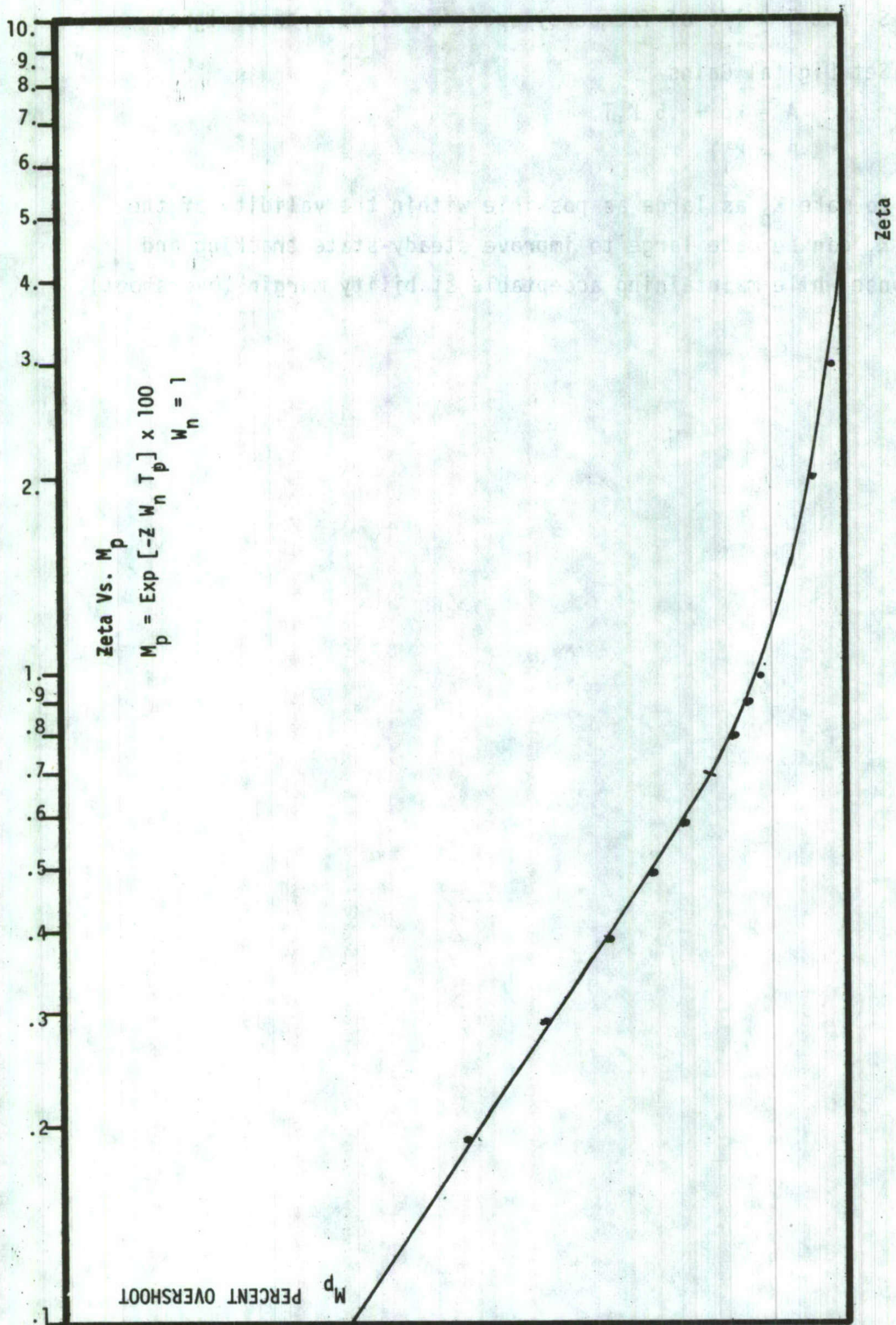


Figure 4. Damping Ratio vs. Peak Overshoot

4. Select $T < 1/6$ of frequency spectrum of reference signal

5. Set Digital Gains

$$A = kp + .5\ K_i T$$

$$B = K_i T$$

The intent is to make K_p as large as possible within the validity of the model so that K_i can be made large to improve steady-state tracking and speed of response while maintaining acceptable stability margin (overshoot).

SECTION IV

DIGITAL LOAD CONTROLLER SOFTWARE IMPLEMENTATION

The flowchart for the software module to be called in turn for each Jth channel controller is shown below:

START

Fetch A_j and X_j (t_k)

Fetch R_j (t_k) From Master

Read (Sample) C_j (t_k)

Compute $E_j (t_k) = R_j (t_k) - C_j (t_k)$

Compute $M_j (t_k) = X_j (t_k) + A_j E_j (t_k)$

Output M_j (t_k)

Fetch B_j (t_k)

Compute $Xj (t_k+1) = X_j (t_k) + B_j E_j (t_k)$

Store X_j $(t_k + 1)$ in X (t_k)

RETURN

The intent of the module design is to minimize the computational transmission time delay by performing the minimum number of computations and operations between the time jth load cell is sampled, C_j (t_k), and the time the resulting command, M_j (t_k), is transmitted to the jth hydraulic jack system (only the jth load cell is sampled during the jth iteration of the loop).

Computer program S.FOR performs the above algorithm for 12 channels. S.FOR is located in the slave machine of the master/slave computer configuration. The program listing is in Appendix A.

SECTION V

LOAD CONTROLLER EXPERIMENTAL VERIFICATION

This section describes experiments performed on a test stand to verify that the digital controller behaves like an analog second-order Proportional-Integral controller in accordance with the analysis of Section III. The verification was done by determining the peak overshoot levels and settling times of the digital controller and comparing them to the analytical predictions of Section III.

The first assumption that overall test stand gain K (1/sec) approximates the product $K_A K_L$ and is independent of the drive signal (reference Section II) was verified with 12 open-loop runs. A strip chart recording of a typical run is shown in Figure 5. The product $K_A K_L = K$ is approximated by the slope of output of load cell (counts/sec.) divided by steady hydraulic jack drive (counts). For example, for Run 5 in Figure 5 a chart speed of 125mm/sec, a vertical chart scale of 400 counts/mm and a steady drive of 8000 counts (10 volts) produced a gain $K = 3.2689 \left[\frac{45\text{mm x } 400 \text{ counts/in}}{83/125\text{mm/sec}} \div 8000 \text{ counts}\right]$. Table 1 lists the results of the 12 runs.

The next set of experiments were made with proportional control only. Strip chart recordings of runs 8, 9, 10 and 11 are shown in Figure 6. Instability and overload trip occurs at A=2 (A is the digital equivalent of analog K_p when B = 0). The non-zero steady-state errors proportional to 1/A and tendency towards instability* for increasing A are characteristic for an analog proportional controller.

---

*A notch filter centered in the lowest measured resonance frequency extended the region of stability for the proportional controller. For notch filter design, see Section VI.

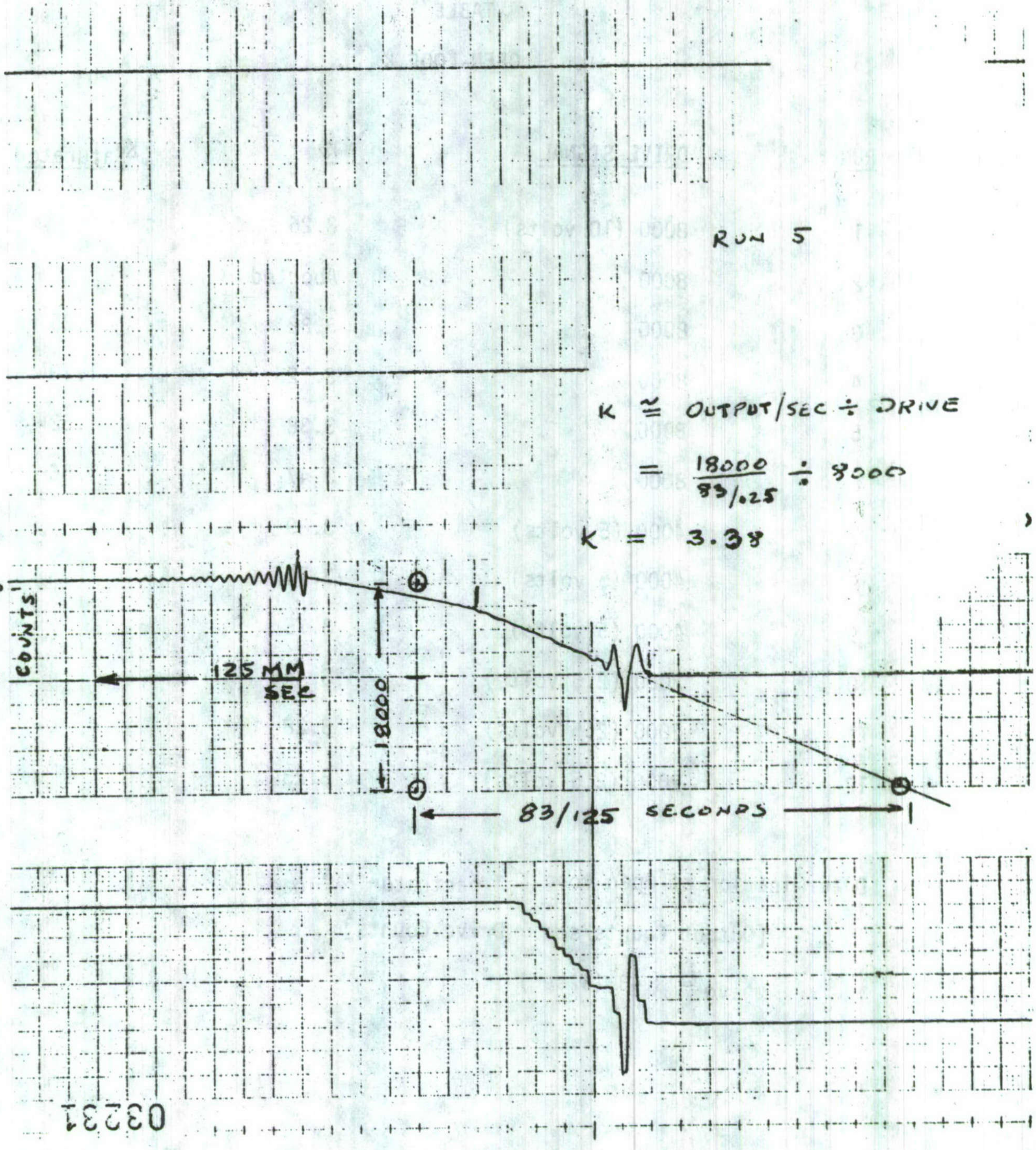


Figure 5. Open-Loop Gain K of Test Stand

TABLE 1

OPEN-LOOP K

RUN	DRIVE SIGNAL	K	$K_{Saturated}$
1	8000 (10 volts)	3.26	
2	8000	Aborted	
3	8000	3.32	1.25
4	8000	3.125	1.28
5	8000	3.38	1.62
6	8000	3.47	1.56
7	4000 (5 volts)	3.89	
8	4000 (5 volts)	3.75	
9	4000 (5 volts)	3.75	
10	2000 (2.5 volts)	3.448	
11	2000 (2.5 volts)	3.28	
12	2000 (2.5 volts)	3.23	

Determination of Open-Loop $K = K_{actuator} \times K_{load}$

[Output Counts/sec ÷ Drive Counts]

$K_{avg} = 3.43$ (1/sec)

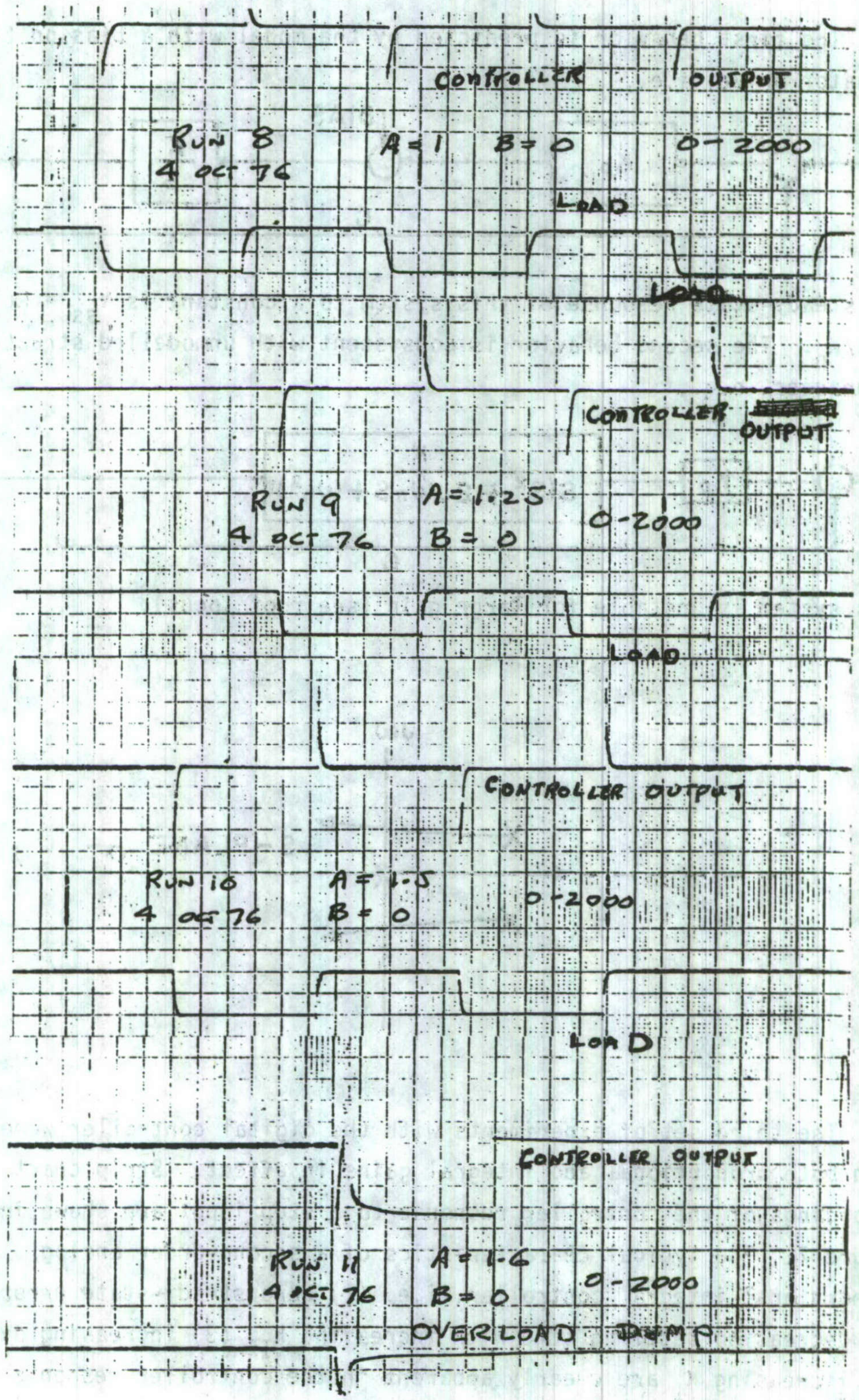
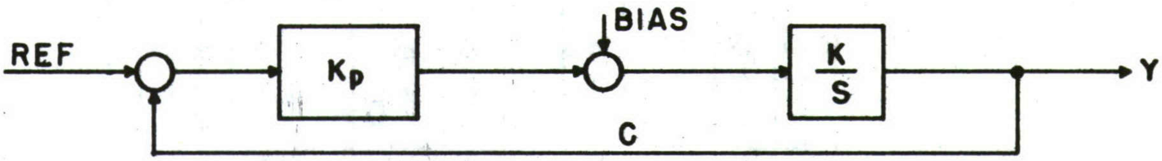
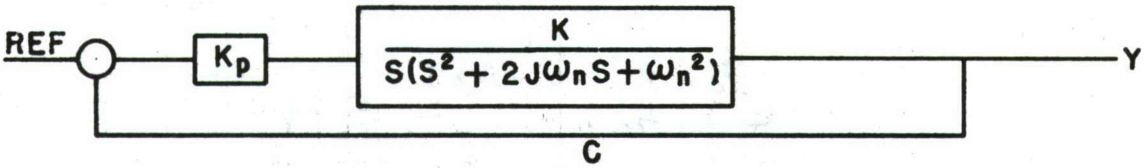


Figure 6. Overload Dump

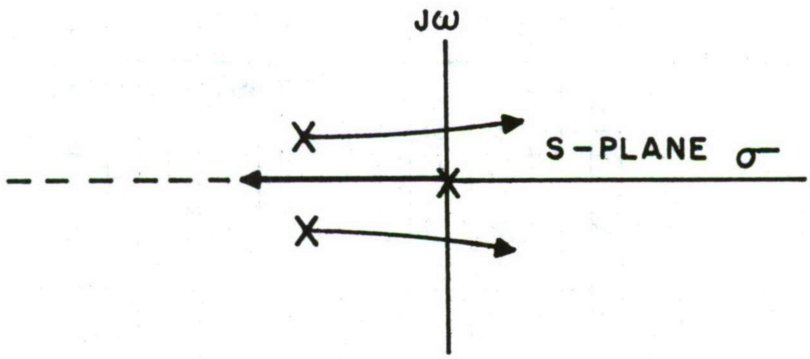
The first behavior is predicted by the model with a bias on the actuator drive, i.e.,



The steady-state response of this system to a constant is Y_{ss} = Ref + Bias/K_p. The second behavior is consistent with unmodelled structural resonances, e.q.,



This system is unstable for large gain (see root locus).



The third set of experiments with the digital controller were made with both proportional and integral gains in effect. Strip chart recordings of the controller response to a step input are shown in Figure 7. The typical characteristics of a second-order analog proportional-integral controller, i.e. (1) zero steady-state error, (2) decreasing settling time with increasing K_I, (3) increasing overshoot for increasing K_I are clearly apparent in the controller response curves.

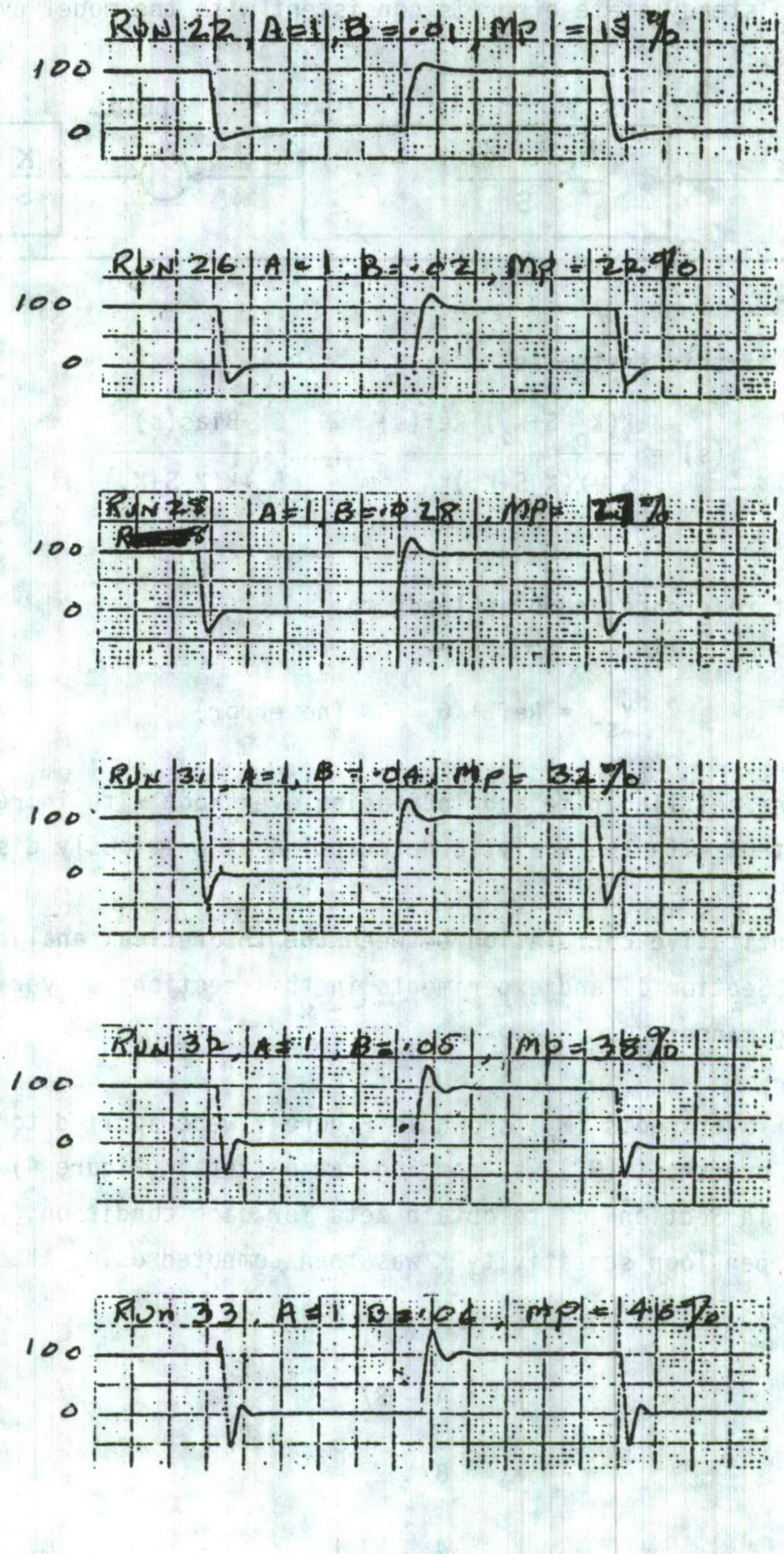
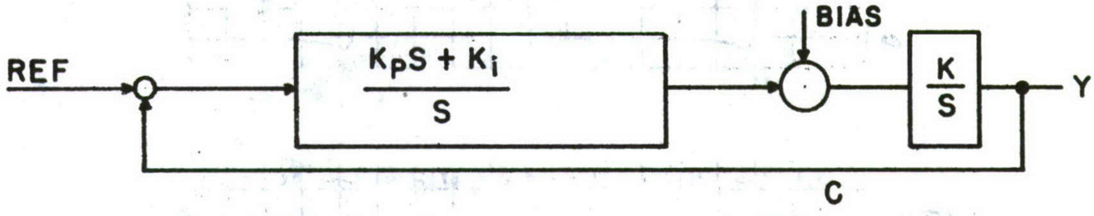


Figure 7. Controller Response to Step Input

The zero steady-state error is consistent with the model even with the bias, i.e.:



The response of this system is

$$Y(s) = \frac{K(K_p\ S+K_i)\ Ref(s)}{S^2+K(K_pS+K_i)} + \frac{SK\ Bias(s)}{S^2+K(K_pS+K_i)}$$

so a constant reference and bias leads to

$$Y_{ss} = Ref + 0 \qquad \text{(no error)}$$

The decreasing settling time and increasing overshoot with increasing K_i is consistent with the analytic description as previously discussed.

The quantitative correlation between the theoretical analysis presented in Section II and experiments in this section was verified in the following manner.

The peak overshoots (M_p) shown in Figure 7 were applied to the plot of peak overshoot (M_p) vs. damping ratio (zeta) (Figure 4) developed analytically in Section III to obtain zeta for each condition. The theoretical open-loop sensitivity K was then computed using the relationships:

$$K_p = A - B/2$$

$$K_i = B/t$$

$$K = 4\zeta^2\ KI/K_p^{\ 2}$$

The results of this computation are shown in Table 2 in comparison with the experimentally determined K described in the first part of this section.

These were not closely controlled experiments. However, the correlation is close enough to verify that the assumptions and approximations made in Sections II and III and the software implementation of the Digital Proportional-Integral controller, Section IV, are valid.

TABLE 2

EXPERIMENTAL K VS. COMPUTED K

DIGITAL A	GAINS B	ANALOG K_p	GAINS K_I	EXPERIMENTAL M_p%	FIG. 4 ζ	CALCULATED K	EXPERIMENTAL K_{Avg}
1	.01	.995	1	15	.92	3.4197	3.43
1	.02	.99	2	22	.66	3.555	3.43
1	.03	.986	2.8	27	.55	3.484	3.43
1	.04	.980	4	32	.47	3.68	3.43
1	.05	.975	5	37	.39	3.2	3.43
1	.06	.97	6	43	.35	3.125	3.43

$K_{calculated} = 4\zeta^2 K_I / K_p^2$

SECTION VI

RESONANCE CONTROL WITH NOTCH FILTERS

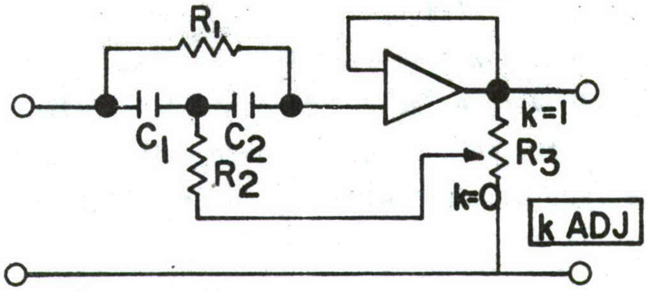
Unexpected vibrations and instabilities in the three channel test stand led to the insertion of notch filters[3] in the output of the digital simulator as a means of suppressing resonant frequencies inherent in the simulated structure. See Figure 12.

Two notch filter designs (Figures 8, 9, 10 and 11) were considered. The notch filter of Figure 8 was implemented, because of simpler construction, and proved very effective in controlling structure resonance for single channels. Fifty percent higher gain settings and ten times the integration rate were possible with the notch filter in the circuit.

The interactions resulting from multi-channel notch filters were not fully understood or investigated.

The full-scale wing-carry-through structure fatigue test was operated without notch filters in the control loop.

[3] Design by Capt Peter Miller, E.E. Dept., AFIT.

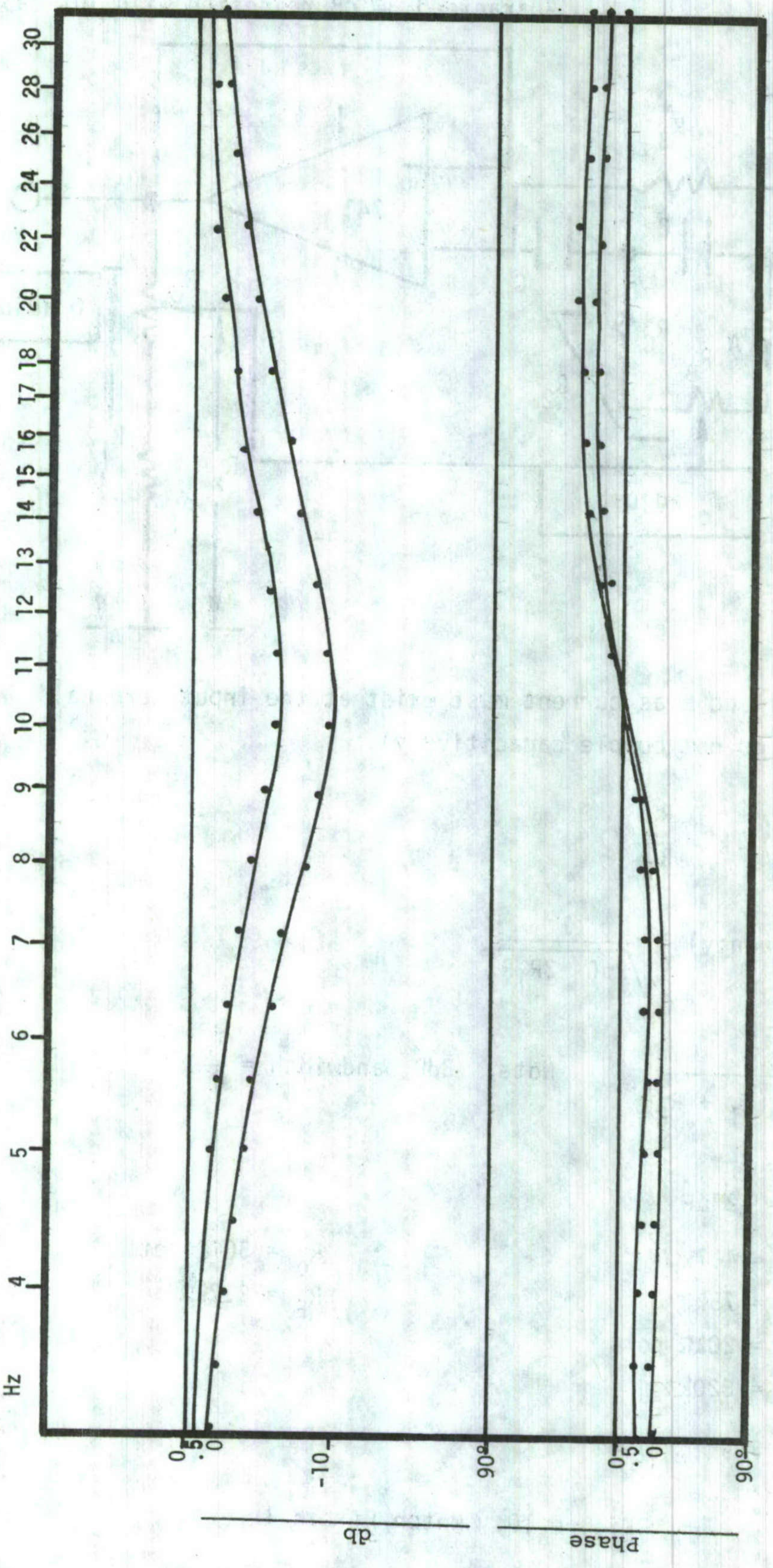


$R_1 = 220\,k\Omega$
$R_2 = 6.8\,k\Omega$
$C_1 = 1.8\,\mu F$
$C_2 = .1\,\mu F$
$R_3 = 1\,k\Omega$

$$WO = \frac{1}{\sqrt{R_1 R_2 C_1 C_2}}$$

$$\boxed{\text{GAIN @ NOTCH} = \frac{1 + C_1/C_2}{1 + C_1/C_2 + R_1/R_2}}$$

Figure 8. Notch Filter A

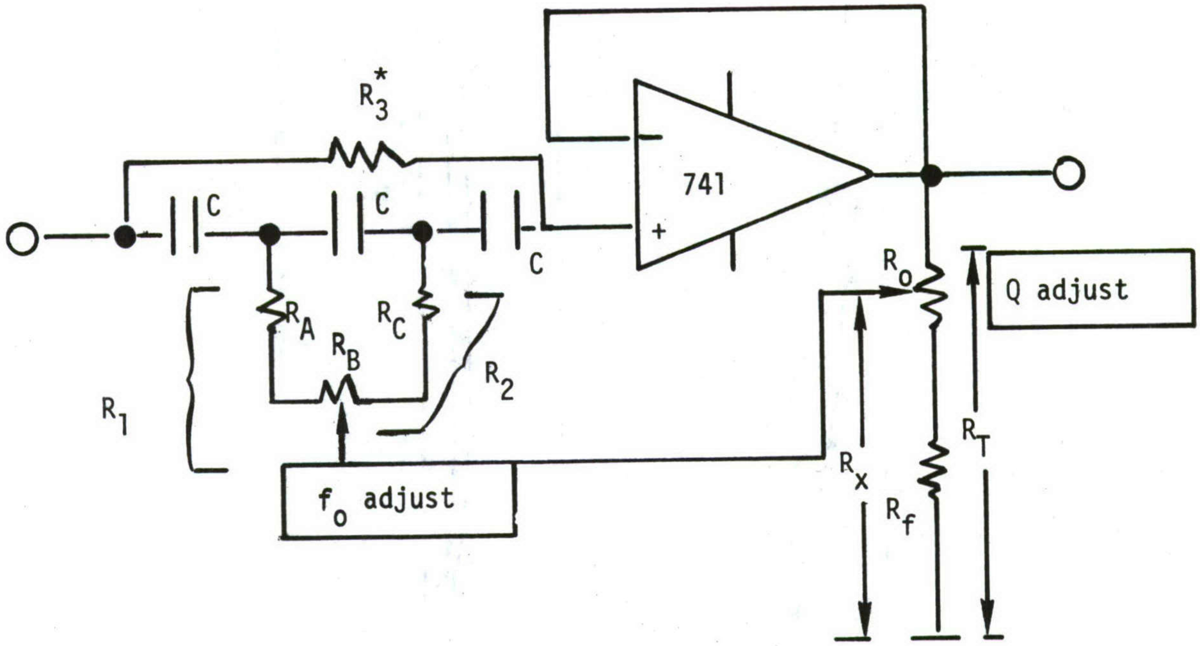


K = 1 is All Pass

$\frac{R1}{R2} = 33 \quad \frac{C1}{C2} = 15$

Figure 9. Frequency Response of Notch Filter A

Guaranteed 30 dB rejection with 10% capacitor



Note: A small dc bias current must exist at the input terminal i.e., do not couple capacitively).

$*R_3 = 6(R_1 + R_2)$; $k = \frac{R_x}{R_T}$

f_o (notch frequency) $\cong \dfrac{1}{\sqrt{2\pi C \quad 3R_1R_2}}$

$Q \cong \dfrac{\sqrt{R_1R_2}}{2(1-k)\sqrt{3}\,(R_1 + R_2)}$ Note: 3dB bandwidth $= \dfrac{f_o}{Q}$

Possible: $C = .27\mu F$

$R_A = 4.7k\Omega$ $R_o = 500\Omega$ pot.

$R_C = 75k\Omega$ $R_E = 2.2k\Omega$

$R_B = 20k\Omega$ pot.

$R_3 = 620k\Omega$

Hatch range: 7.9 to 17 Hz

Figure 10. Notch Filter B

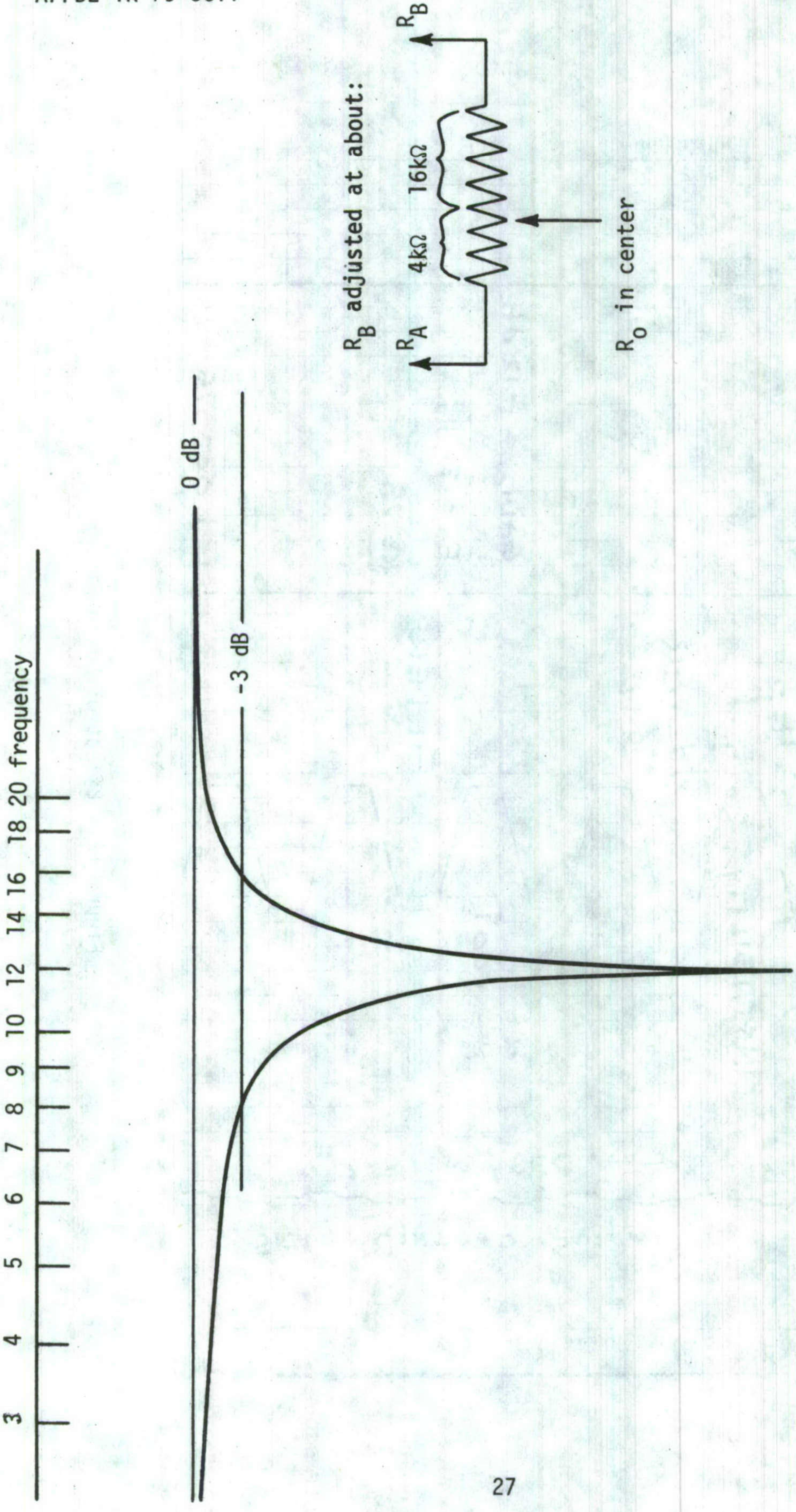


Figure 11. Frequency Response of Notch Filter B

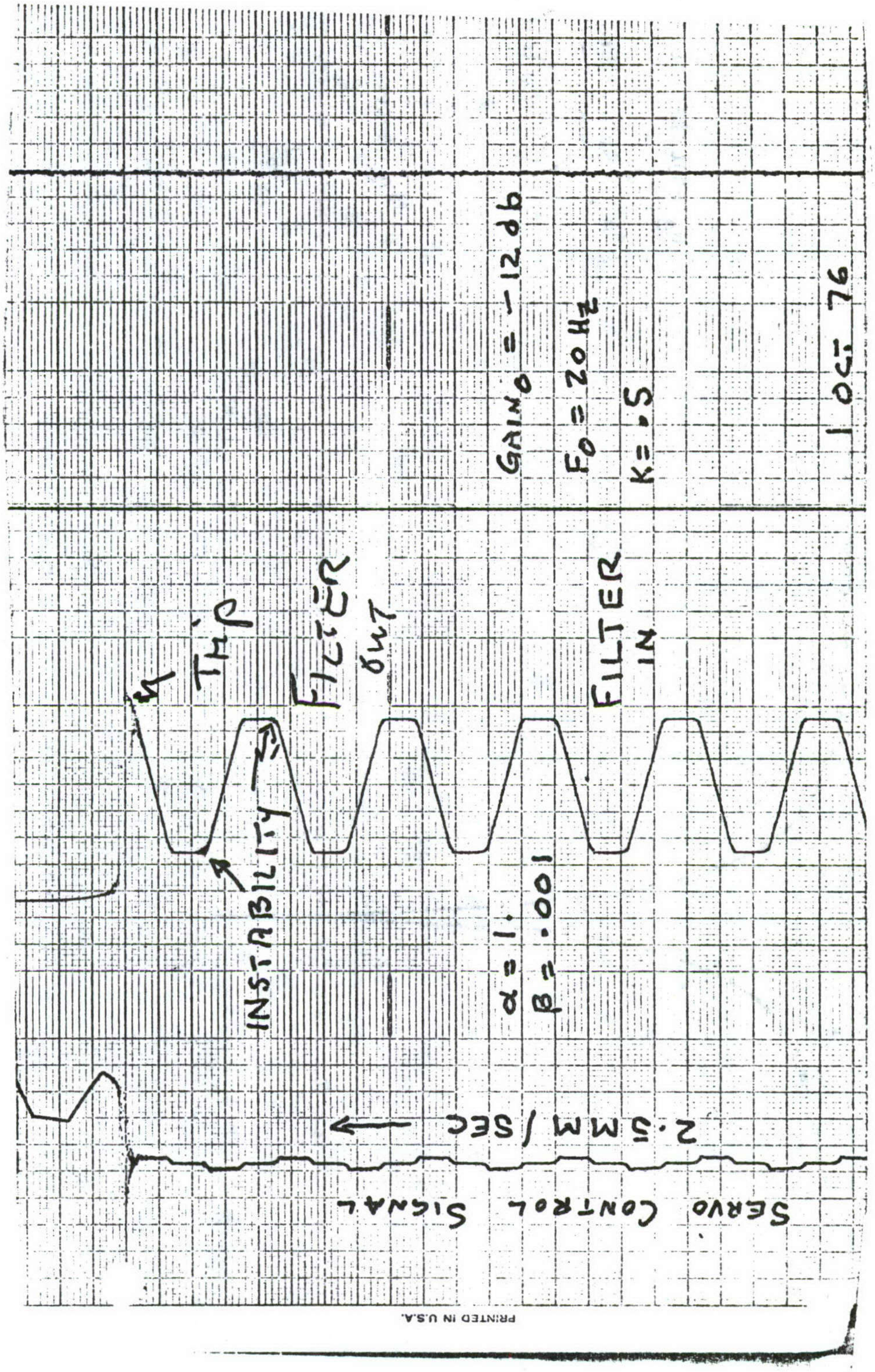


Figure 12. Effect of Notch Filter

SECTION VII

APPLICATION TO FULL-SCALE TESTING

This section describes the successful application of the digital control system to a full-scale multichannel airframe fatigue test (Advanced Metallic Air Vehicle Structure wing carry through structure). This system was substituted for conventional analog controllers during the fourth lifetime of an established fatigue test. The new system was required to match all the existing functional and hardware interfaces for load generation, program check, load check, and start-up and shut-down systems.

The actual transition from one system to another (digital controller vs. analog controller) consisted of switching minicomputer programs and the input plugs to the servo-valves.

Appendix E, Part 4, is a strip chart for a section of the continuously varying load profile applied to the AMAVS test structure on eleven channels.

Noise would appear intermittently on some channels. These were caused by a variety of mechanical malfunctions considered common in large scale fatigue tests, e.g., structural resonances, servo-valve characteristic changes, or over adjustment of gain controls. The interaction between the DDC and dead weight control systems would sometimes cause problems. Cycling speeds at the conclusion of the test were approximately the same as under analog control. At no time were peak overshoots allowed to exceed five percent of applied loads on most channels and ten percent on two of the eleven channels. The best performance of the digital controllers was not as good as the analog controllers, but this is attributed to the lack of experience in setting up large systems and reluctance to experiment with the control system on an expensive test article.

The results of this application were better than expected and are the basis for continuing work in this area.

The balance of this section is a detailed description of hardware and software used to implement DDC on the AMAVS fatigue test.

HARDWARE

An overview of the AMAVS fatigue test operating system is presented in Figure 13. In the arrangement shown, the slave minicomputer is dedicated to the task of functioning as multichannel digital proportional-integral controller.

The master minicomputer generates the load spectrum, transmits reference valves to the slave via high-speed link, and satisfies background requests to control test operation and display systems.

This arrangement permits the slave to operate at the fastest possible sampling speeds, while preventing inadvertent shifts in sampling times by modifications to the function generating program or other program changes imposed by safety or test operating personnel.

The master minicomputer is equipped with 28K, 16-Bit word core memory, with 1.2 microseconds cycle time and an auxiliary 1.2-million-word disk memory system. The master communicates with the slave via a 500,000 words-per-second direct-memory access link. Other peripherals are a digital link with an independent minicomputer which verifies the program levels being output by the master minicomputer and 12 digital-to-analog converters to furnish reference loads to an analog redundant overload dump system. An alphanumeric Cathode Ray Terminal (CRT) display and keyboard provide operator communication with the system.

The Slave minicomputer is similar to the Master, but had 16K word core memory. Reference values for closed loop control were furnished to the Slave by the Master via link. Feedback values obtained from load cells were collected by a 64-channel low-level signal acquisition and multiplexing system (RTP). Conversion frequency is 19KHz. Individual channel signal conditioners and preamps provide a continuously adjustable

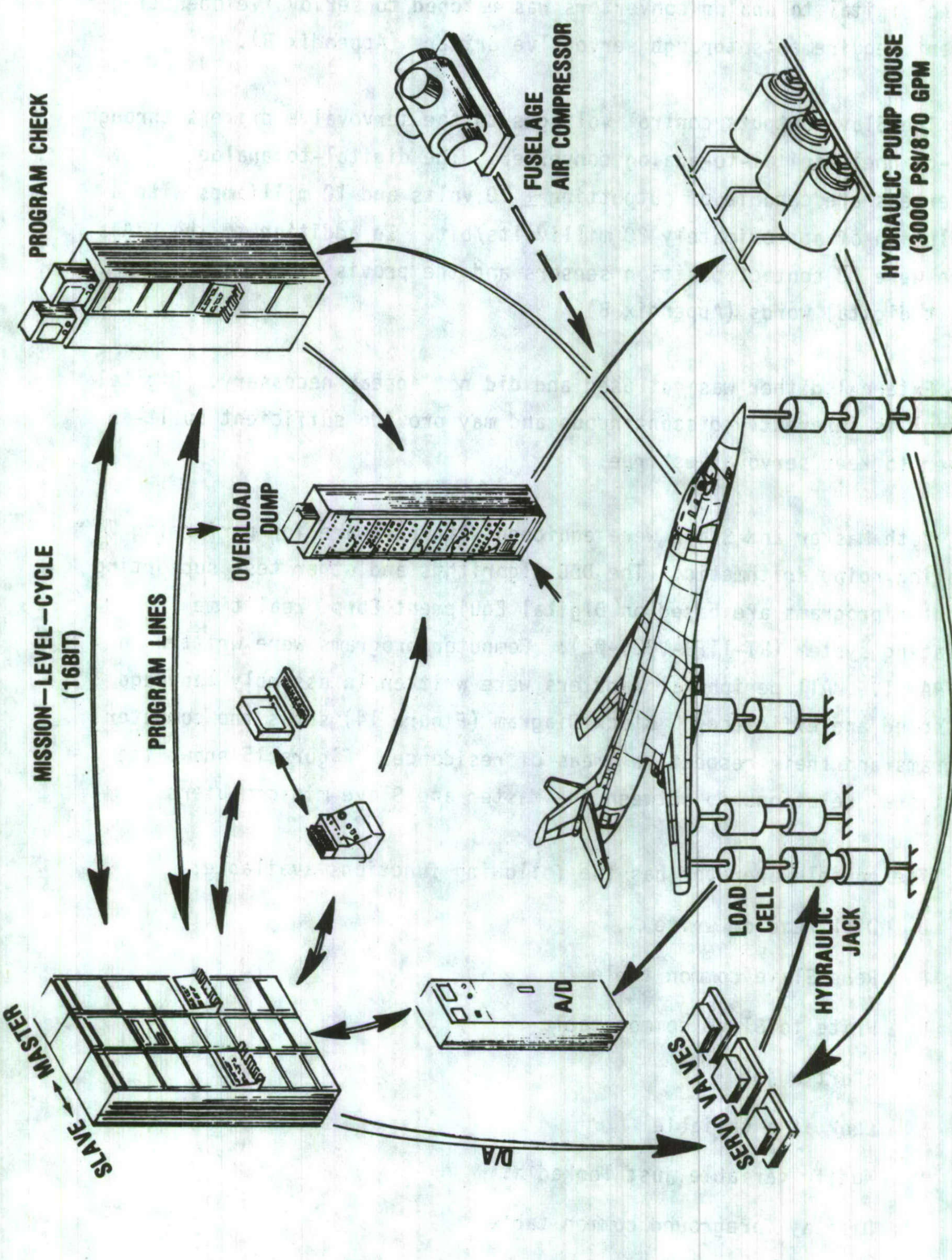


Figure 13. Overview of AMAVS Minicomputer Systems

gain of 1 to 1000. Low pass input filters are provided. Voltage output of the digital to analog converters was matched to servovalve operating current requirements through servovalve drivers (Appendix B).

The Slave outputs control voltages to the servovalve drivers through a 32-channel digital-to-analog converter. The digital-to-analog converters are capable of outputting $\pm$ 10 volts and 10 milliamps with a resolution of approximately 20 millivolts/bit. In addition to the D/A's there were 32 contact position sensors and the provision to output two 16-bit digital words (Appendix B).

External dither was not used and did not appear necessary. Digital control is inherently discontinuous and may provide sufficient built-in dither to keep servovalves free.

Both Master and Slave were equipped with hardware for performing floating point arithmetic. The DDC algorithms and other test supporting computer programs are based on Digital Equipment Corp. real time operating system (RT-11FB-VØ2C-Ø2). Computer programs were written in FORTRAN IV. All peripheral handlers were written in assembly language for speed and efficiency. Block Diagram (Figure 14) shows the computer programs and their respective areas of residence. Figure 15 shows the functional relationship between the Master and Slave minicomputers.

The console operator has the following functions available:

U Disk read or write

R Read Slave common table

W Write to Slave common table

G Start a job

L Look at a variable

M Modify variable just looked at

D Display foreground common table

P Potentiometer control mode

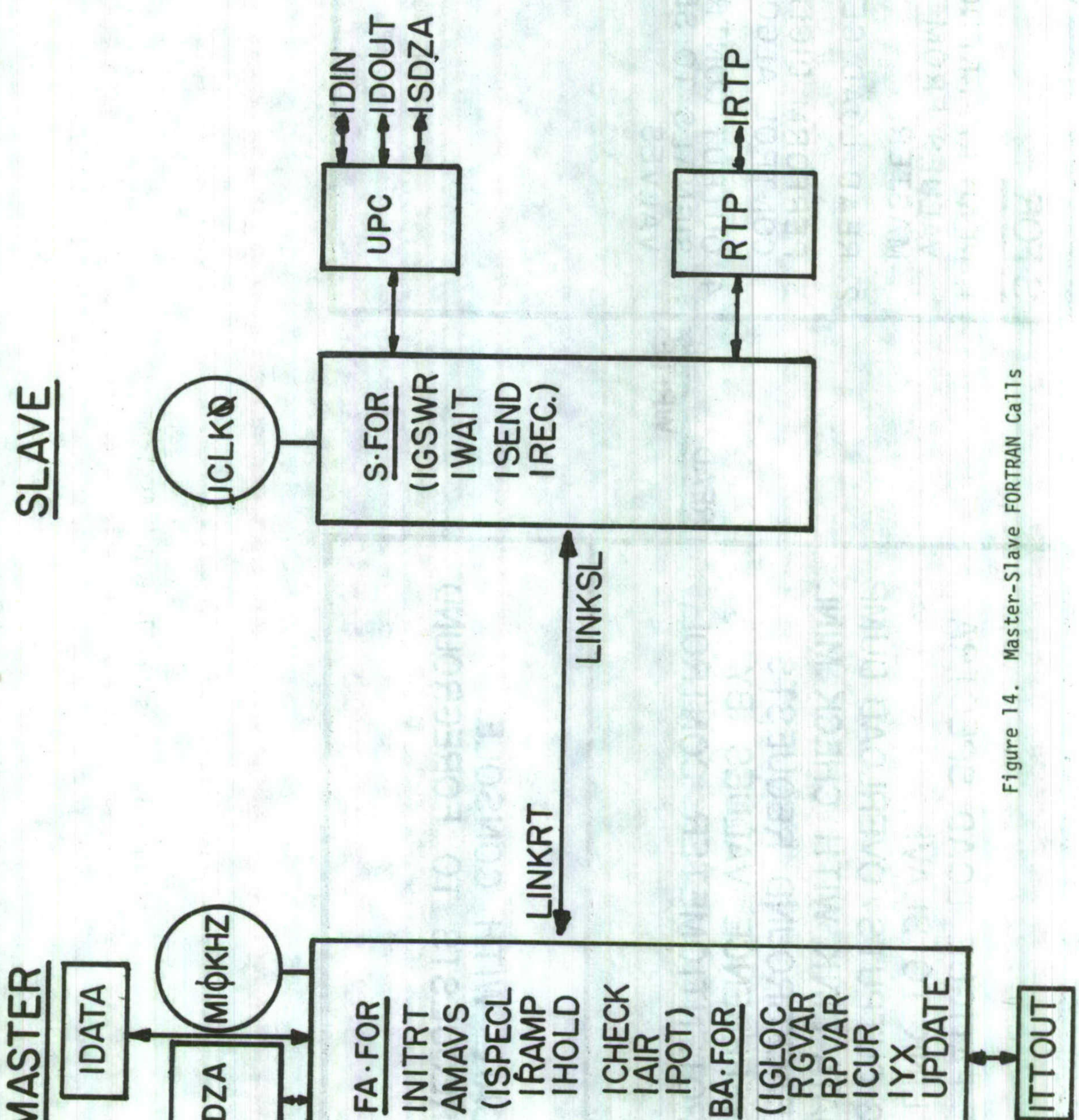


Figure 14. Master-Slave FORTRAN Calls

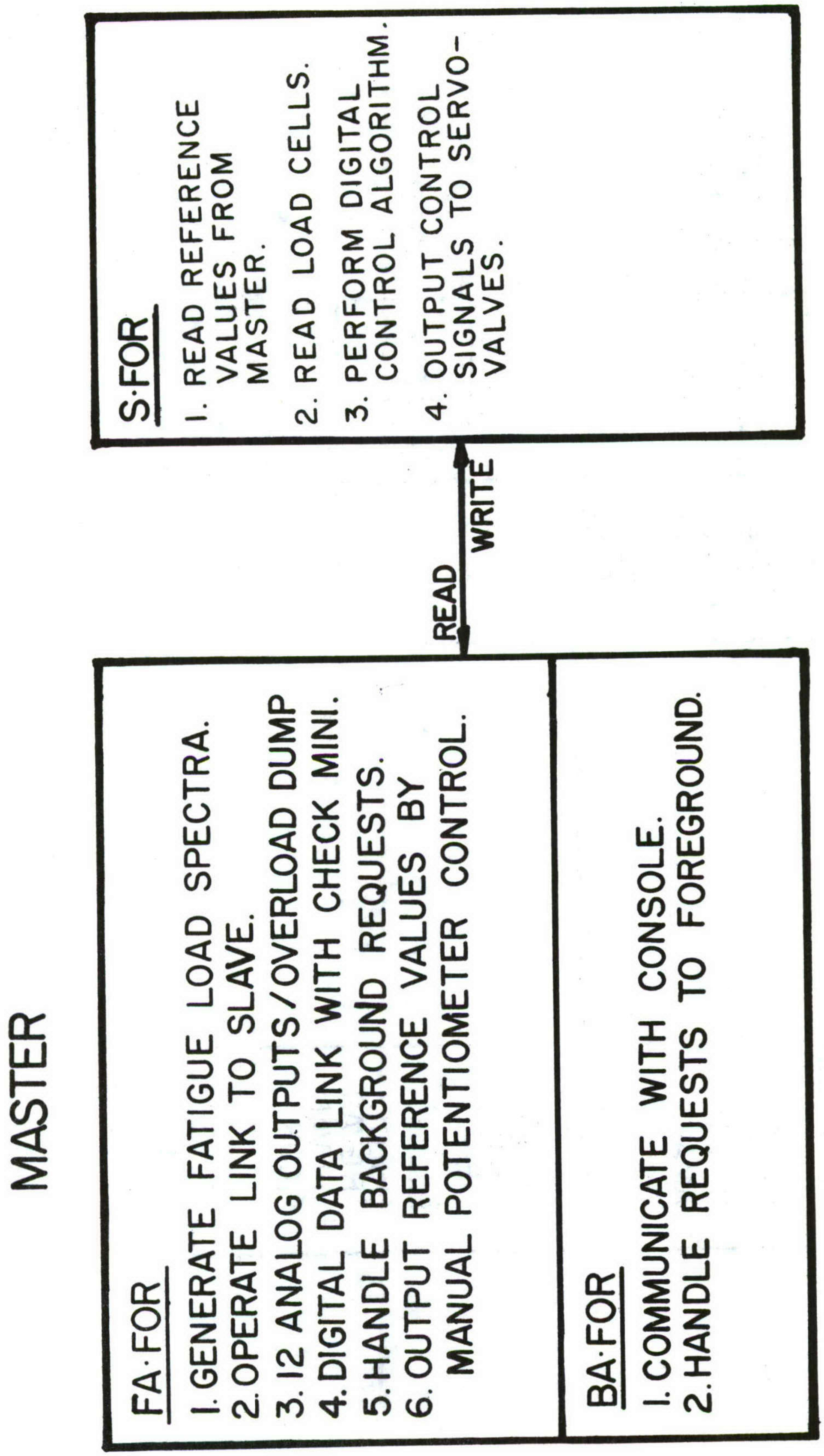


Figure 15. Master-Slave Job Assignments

Z Zero integrators (B's); follow with W

K Kill program (ramp to zero)

H Hold in present position

@ Ramp to reference

DDC TV monitor layout is shown in Figure 16. Printout of common table between the minicomputers is shown in Figure 17.

A digital computer is inherently capable of manipulating a much greater type and variety of information than its analog counterpart. The format for presentation on its CRT display can be varied rather flexibly with FORTRAN format statements. The format shown in Figure 16 was selected as the most useful for this application.

Meaning of code appearing in Status 1 and Status 2 columns are shown in Tables 3 and 4.

Flow charts for Master Programs FA.FOR and BA.FOR, and Slave Program S.FOR are given in the following pages. The complete program listings are included in Appendix A.

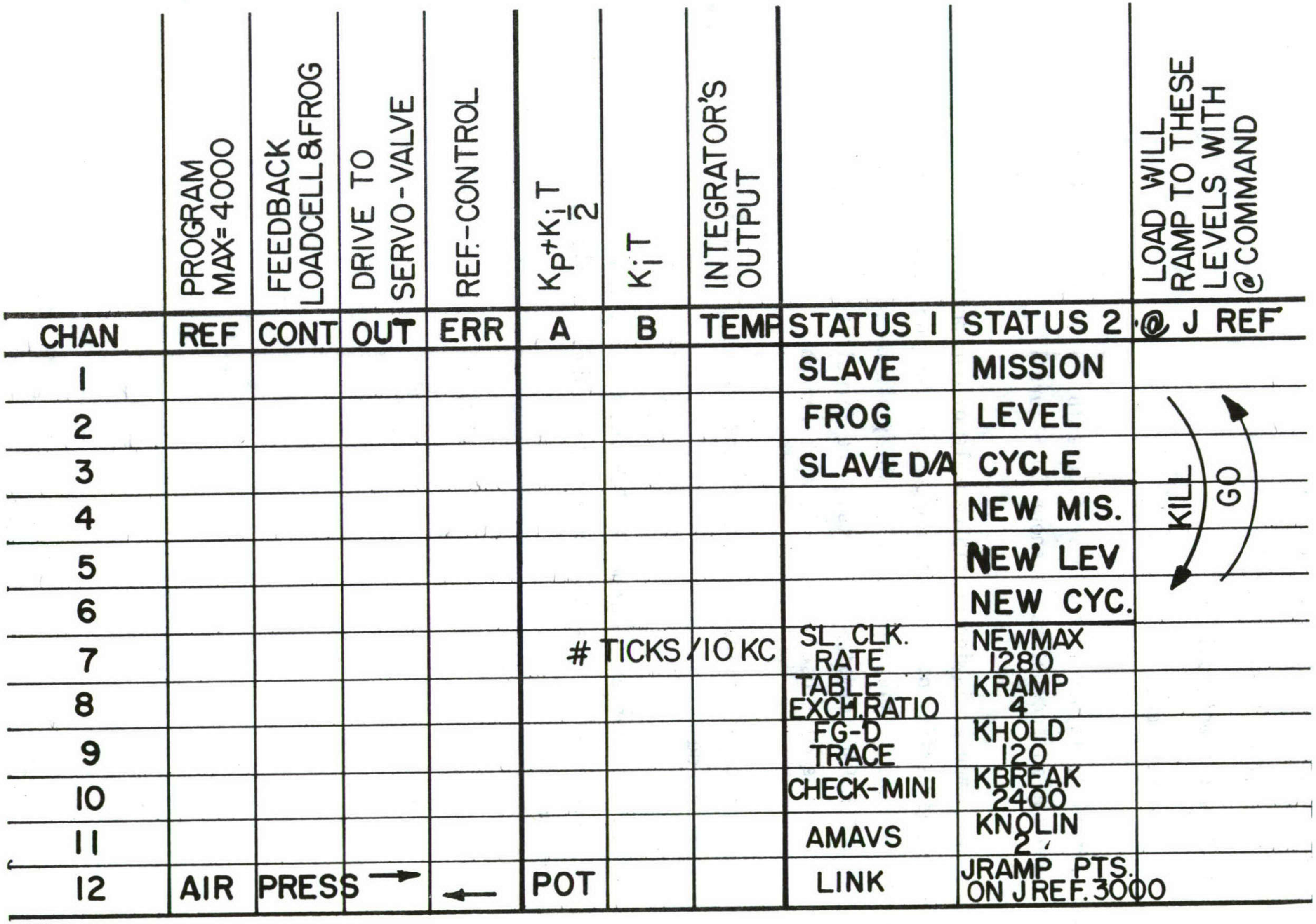


Figure 16. DDC Instrument Panel

CHAN	REF	CONTROL	OUTPUT	ERROR	A	B	TEMPX	STAT-1	STAT-2	@JREF
1	2297	-5	0	5	0.00	0.0000	0.0	0	1234	0
2	1199	-7	0	7	0.00	0.0000	0.0	0	42	0
3	1853	-2	0	2	0.00	0.0000	0.0	0	1	0
4	144	-3	0	3	0.00	0.0000	0.0	0	0	0
5	-153	-7	0	7	0.00	0.0000	0.0	0	0	0
6	-41	-5	0	5	0.00	0.0000	0.0	0	0	0
7	-385	-4	0	4	0.00	0.0000	0.0	100	1280	0
8	-2460	-8	0	8	0.00	0.0000	0.0	0	6	0
9	535	-11	0	11	0.00	0.0000	0.0	0	120	0
10	0	-8	0	8	0.00	0.0000	0.0	1	2800	0
11	-1252	-5	0	5	0.00	0.0000	0.0	1	2	0
12	6000	-3	0	0	0.00	0.0000	0.0	0	3000	0
				IN=	0	0	OUT=	0	0	

FUNCTION

Figure 17. Printout of Common Table in Master

TABLE 3

INSTRUMENT PANEL CODE FOR STATUS 1

CHANNEL	CODE	MEANING
10	0	Run with checkmini
	1	Bypass checkmini
	2	
	3	
11	0	Hold load level
	1	Run AMAVS load spectrum
	2	Ramp to reference loads (effective when not running AMAVS)
	3	Pot control (structure loads follow pot setting)
	-1	Consol kill (ramp loads to zero)
	-2	Hydraulic dump
12	1	Transmit to Slave
	2	Receive from Slave
	3	Transmit to disk
	4	Receive from disk
	5	Transmit In, Iout, IR to Slave
	6	Receive In, Iout, IR from Slave

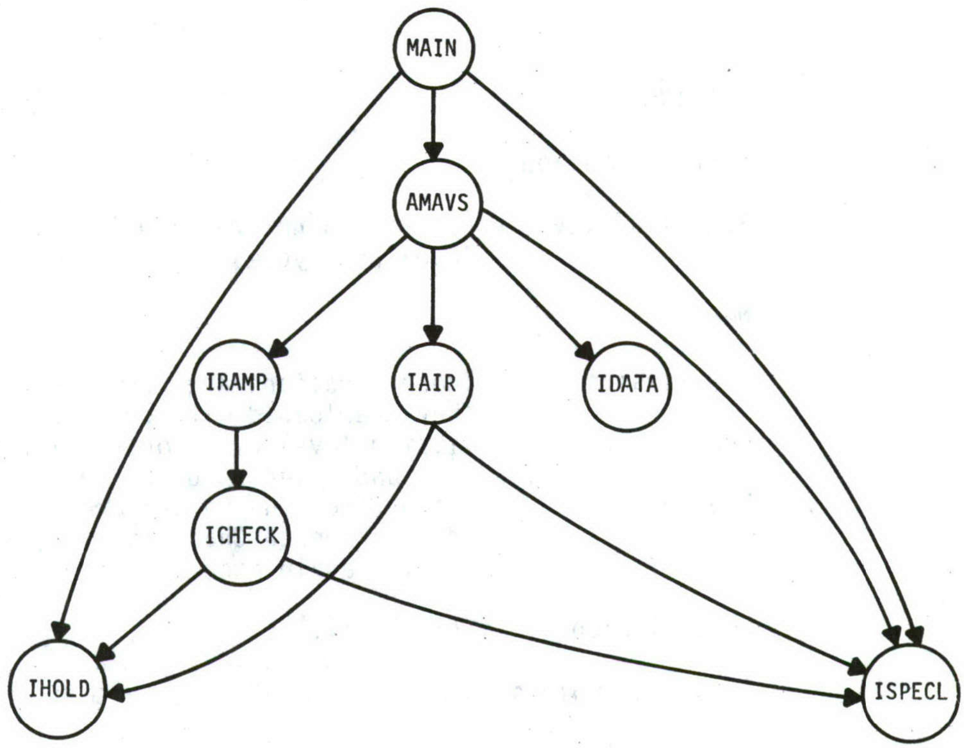
TABLE 4

INSTRUMENT PANEL CODE FOR STATUS 2

CHANNEL	MEANING	
1	Current mission	
2	Next load level - Program is now ramping to this level and cycle	
3	Next cycle	
4	New mission	These locations are normally zero. They are loaded with current mission level cycle on a kill or dump and saved on disk. On restarting, the values are transferred to Channels 1, 2, 3 and 4, 5, 6 again become zeros
5	New level	
6	New cycle	
7	Next mission limit (1280 Max.)	
8	Total No. points in a ramp are divided by this constant to control ramp slope	
9	Hold time (Nx60) (N = seconds)	
10	Ramps with more points than this number may run at a different speed	
11	Multiplier to use with above	
12	No. of points used with @ command to ramp to levels in @ J_{ref} column	

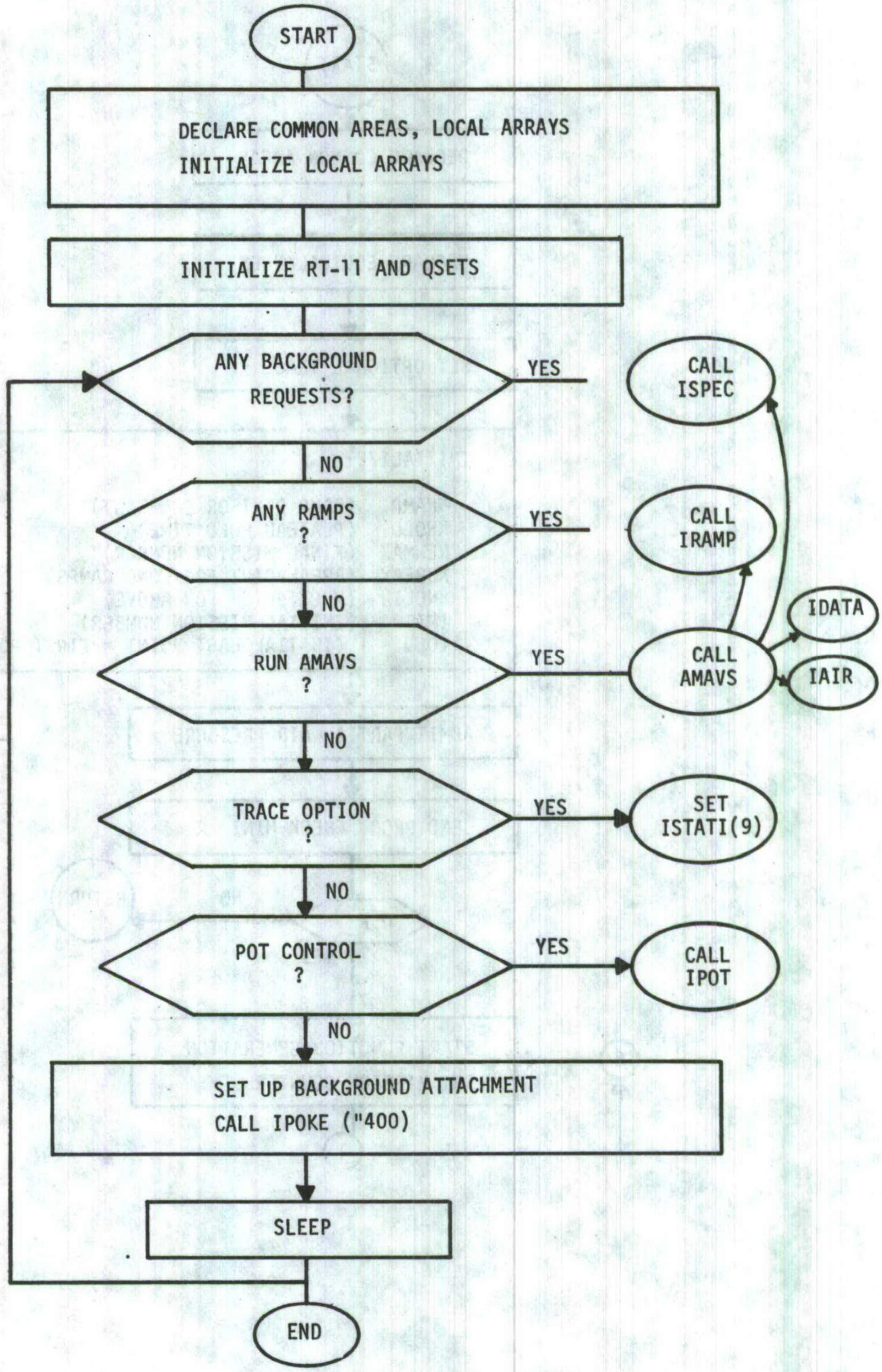
SOFTWARE

FA. FOR (ARCHITECTURE)

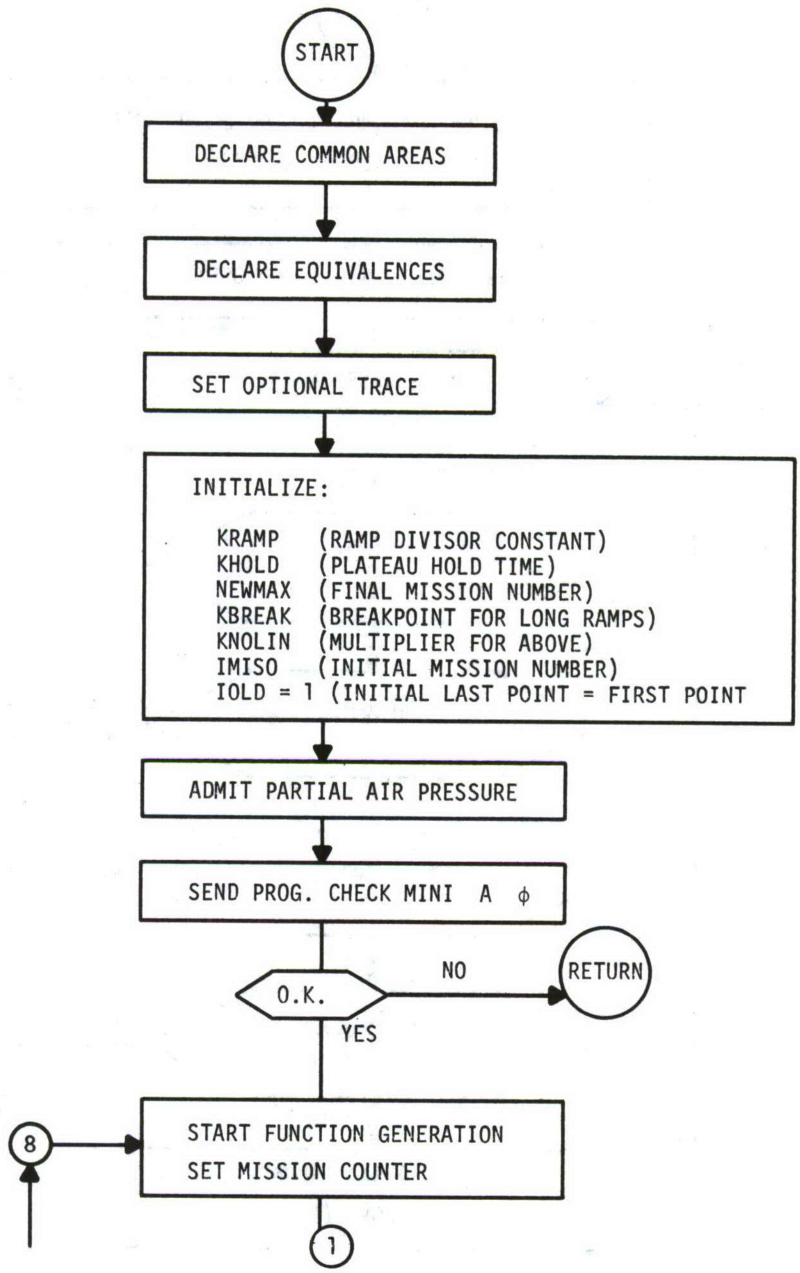


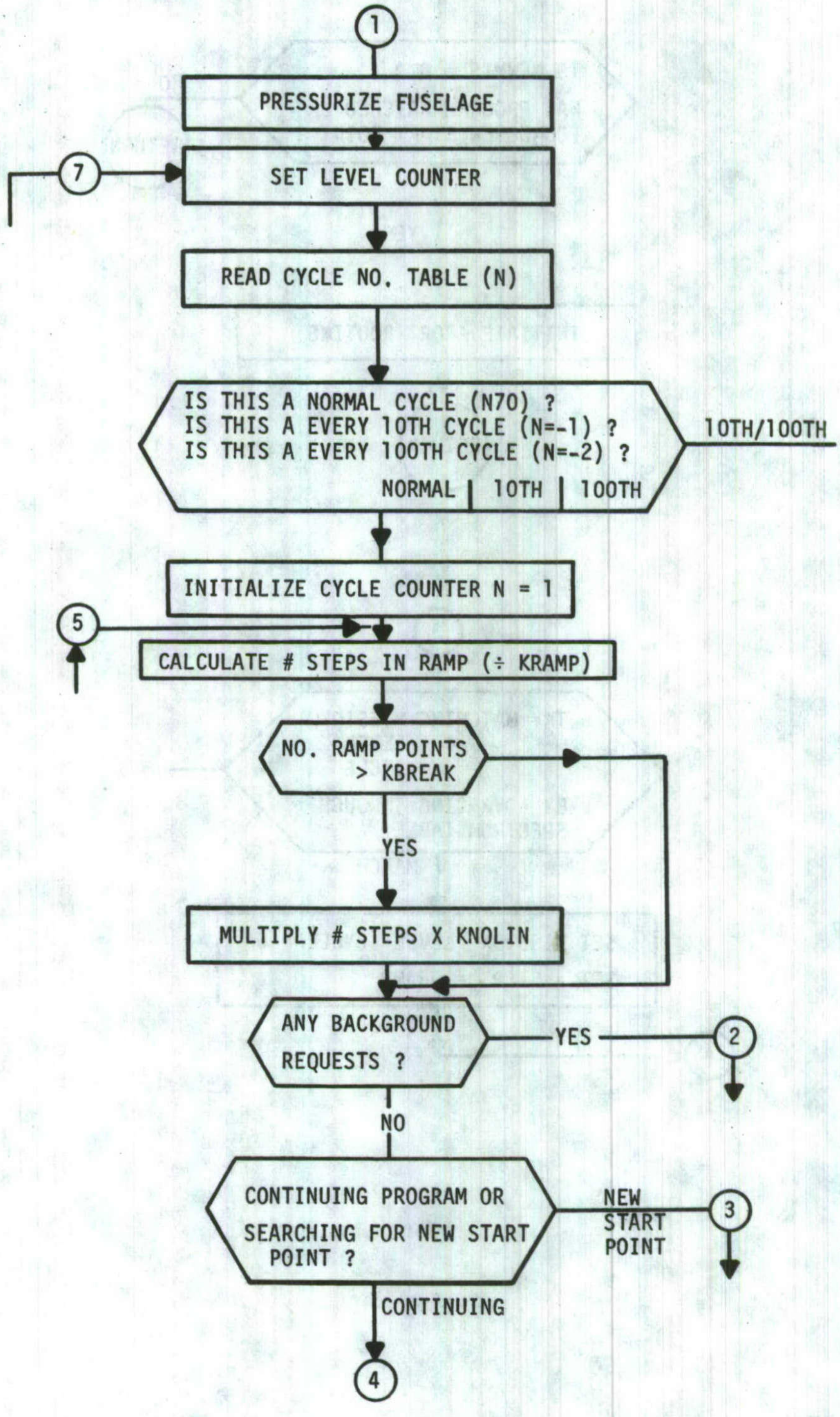
FLOW CHARTS FOR THE MAIN ROUTINE FUNCTIONS AND SUBROUTINES FOLLOW:

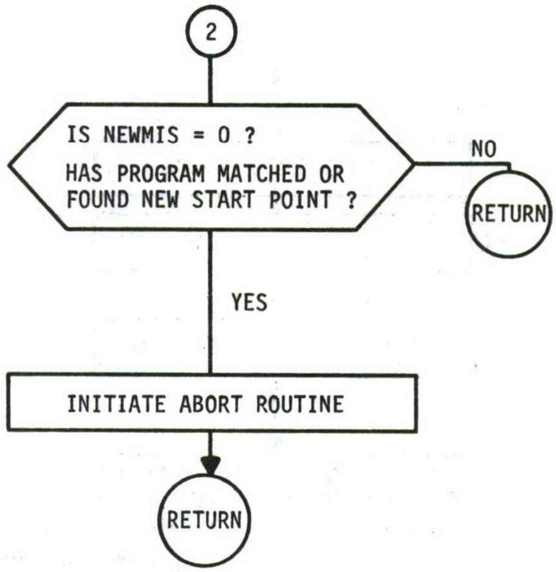
FA. FOR (FLOW CHART)

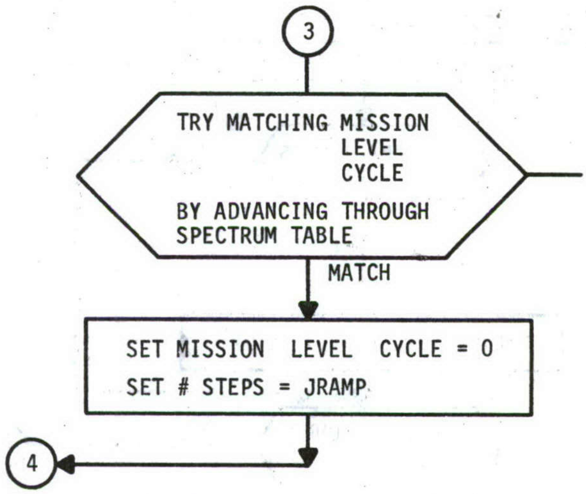


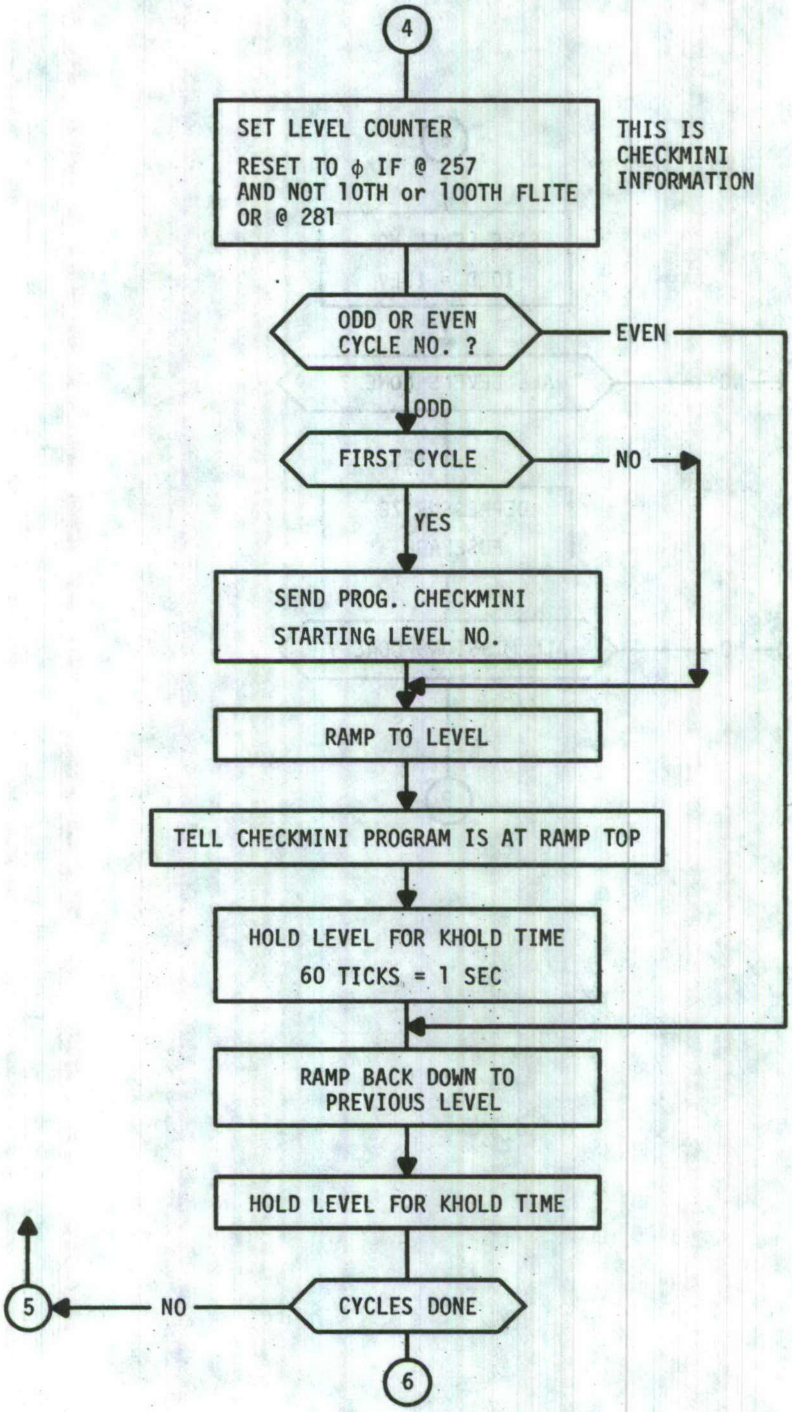
FUNCTION AMAVS

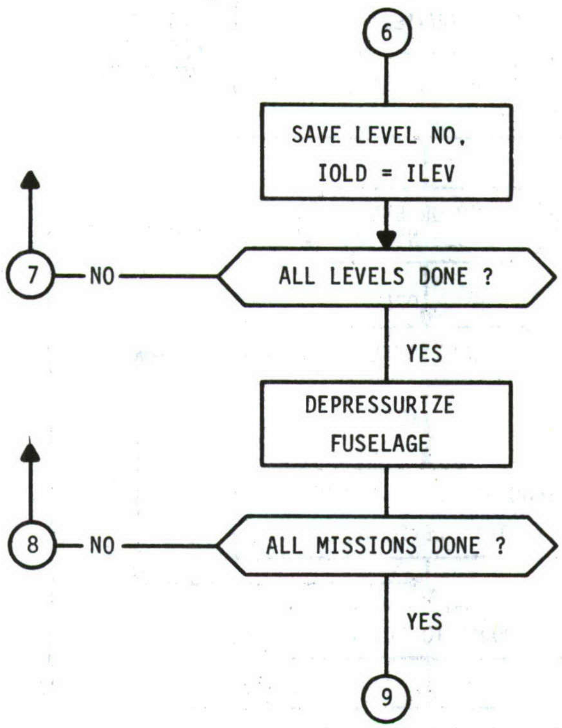


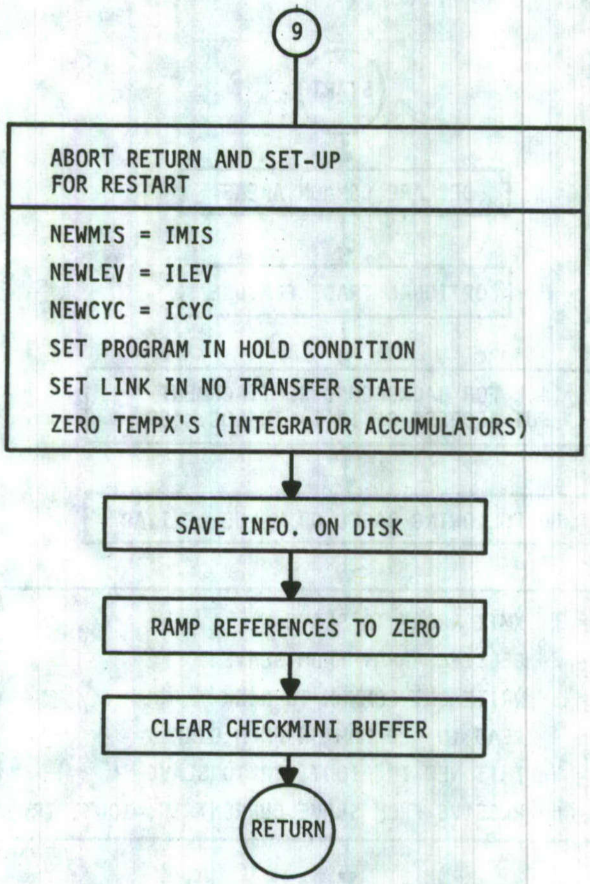

1
PRESSURIZE FUSELAGE
7
SET LEVEL COUNTER
READ CYCLE NO. TABLE (N)
IS THIS A NORMAL CYCLE (N70) ?
IS THIS A EVERY 10TH CYCLE (N=-1) ?
IS THIS A EVERY 100TH CYCLE (N=-2) ?
10TH/100TH
NORMAL
10TH
100TH
INITIALIZE CYCLE COUNTER N = 1
5
CALCULATE # STEPS IN RAMP (÷ KRAMP)
NO. RAMP POINTS > KBREAK
YES
MULTIPLY # STEPS X KNOLIN
ANY BACKGROUND REQUESTS ?
YES
2
NO
CONTINUING PROGRAM OR SEARCHING FOR NEW START POINT ?
NEW START POINT
3
CONTINUING
4


2
IS NEWMIS = 0 ?
HAS PROGRAM MATCHED OR
FOUND NEW START POINT ?
NO
RETURN
YES
INITIATE ABORT ROUTINE
RETURN

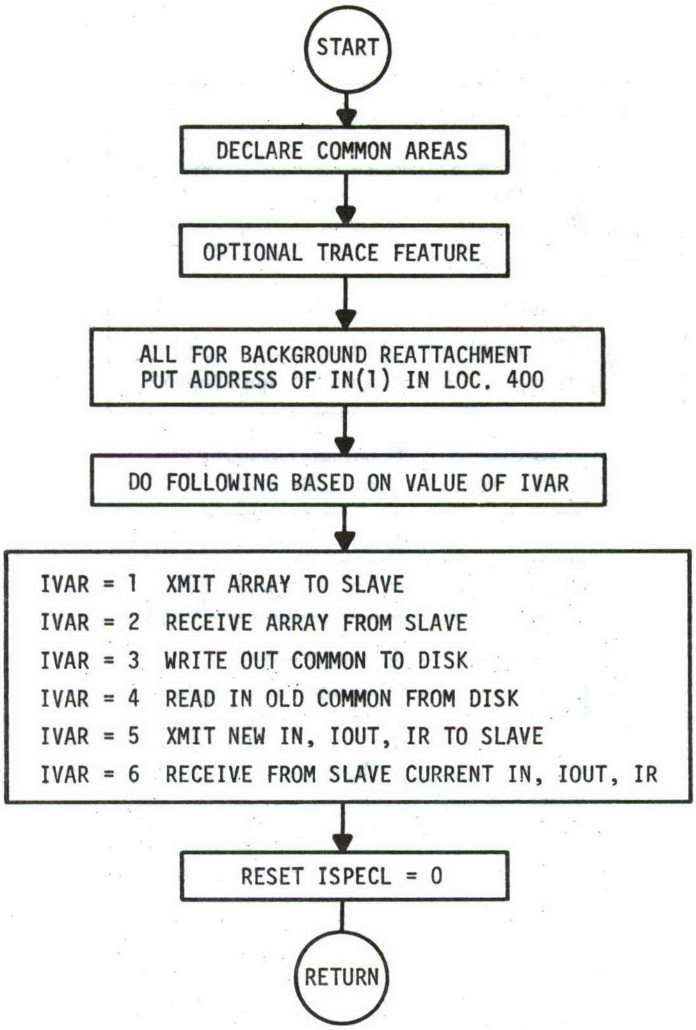

3
TRY MATCHING MISSION
LEVEL
CYCLE
BY ADVANCING THROUGH
SPECTRUM TABLE
MATCH
SET MISSION LEVEL CYCLE = 0
SET # STEPS = JRAMP
4


4
SET LEVEL COUNTER
RESET TO ϕ IF @ 257
AND NOT 10TH or 100TH FLITE
OR @ 281
THIS IS
CHECKMINI
INFORMATION
ODD OR EVEN
CYCLE NO. ?
EVEN
ODD
FIRST CYCLE
NO
YES
SEND PROG. CHECKMINI
STARTING LEVEL NO.
RAMP TO LEVEL
TELL CHECKMINI PROGRAM IS AT RAMP TOP
HOLD LEVEL FOR KHOLD TIME
60 TICKS = 1 SEC
RAMP BACK DOWN TO
PREVIOUS LEVEL
HOLD LEVEL FOR KHOLD TIME
CYCLES DONE
NO
5
6

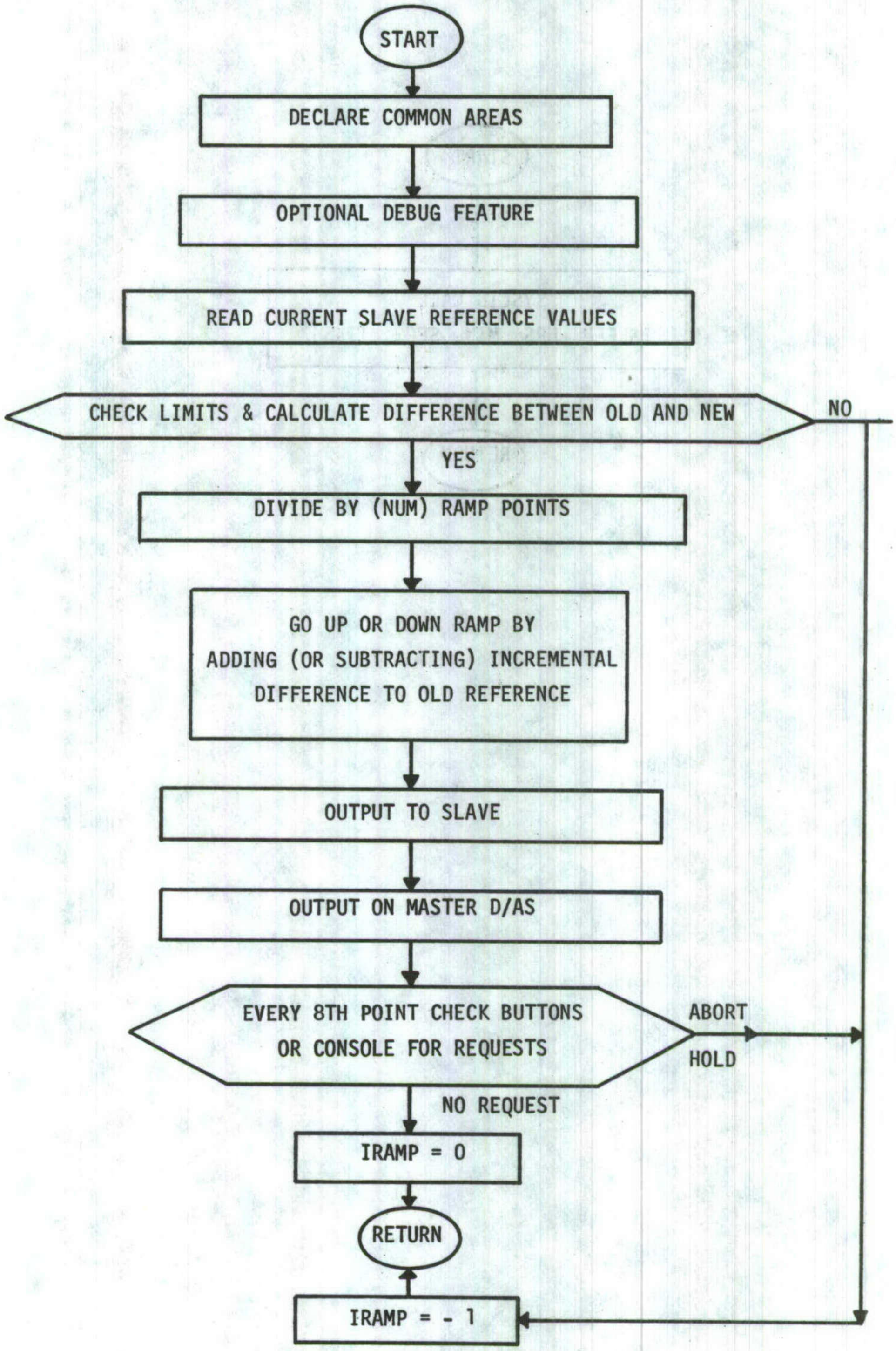

6
SAVE LEVEL NO.
IOLD = ILEV
7
NO
ALL LEVELS DONE ?
YES
DEPRESSURIZE
FUSELAGE
8
NO
ALL MISSIONS DONE ?
YES
9


9
ABORT RETURN AND SET-UP FOR RESTART
NEWMIS = IMIS
NEWLEV = ILEV
NEWCYC = ICYC
SET PROGRAM IN HOLD CONDITION
SET LINK IN NO TRANSFER STATE
ZERO TEMPX'S (INTEGRATOR ACCUMULATORS)
SAVE INFO. ON DISK
RAMP REFERENCES TO ZERO
CLEAR CHECKMINI BUFFER
RETURN

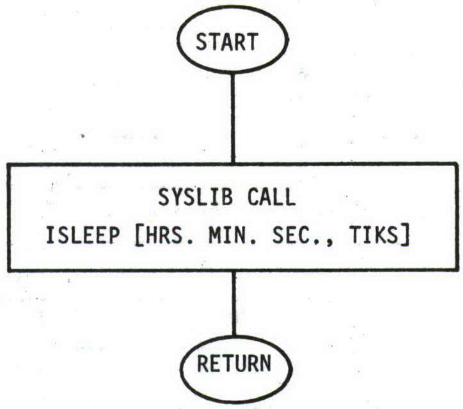
FUNCTION ISPECL (IVAR)



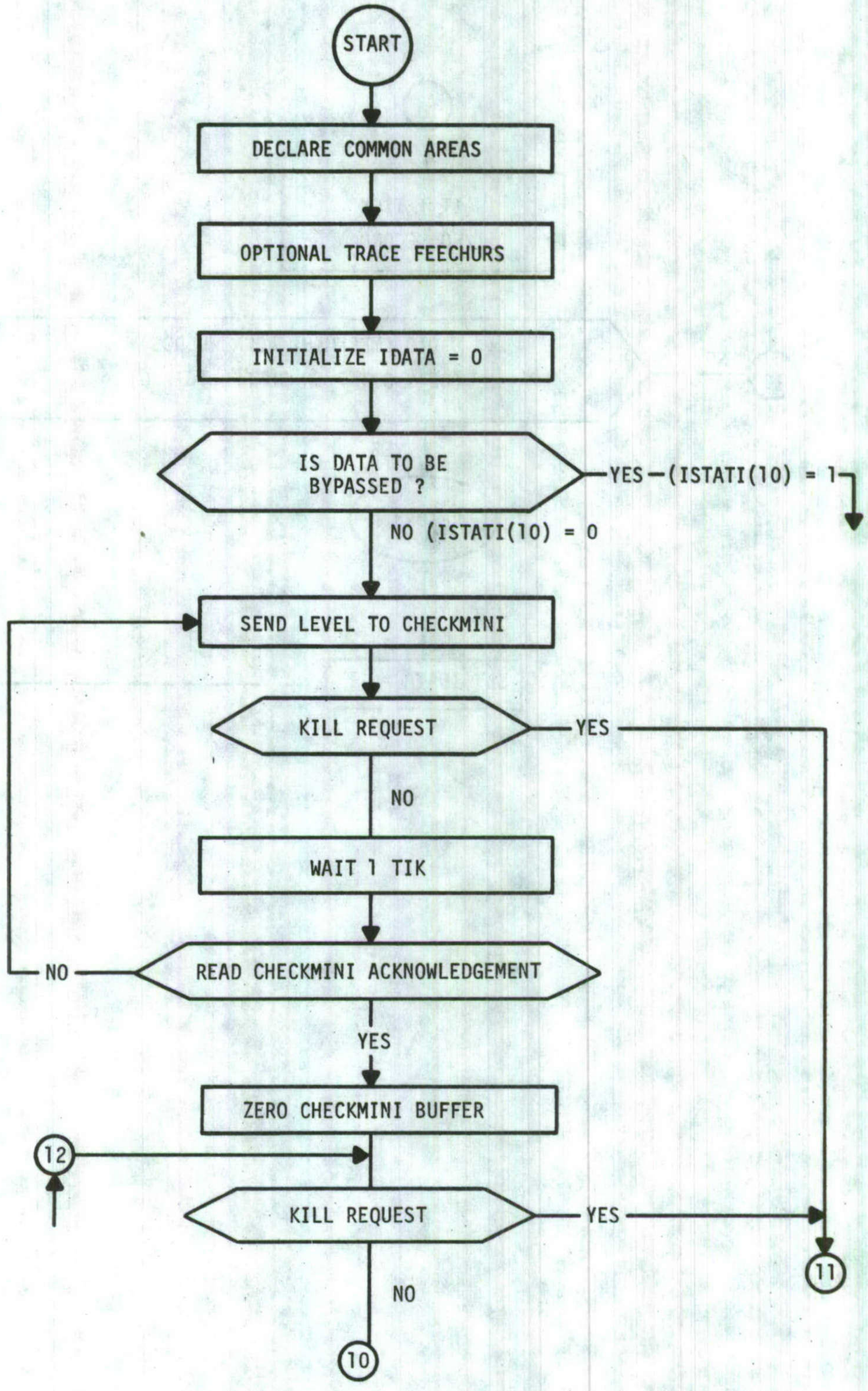
FUNCTION IRAMP (NEW, NUM, ABORT)



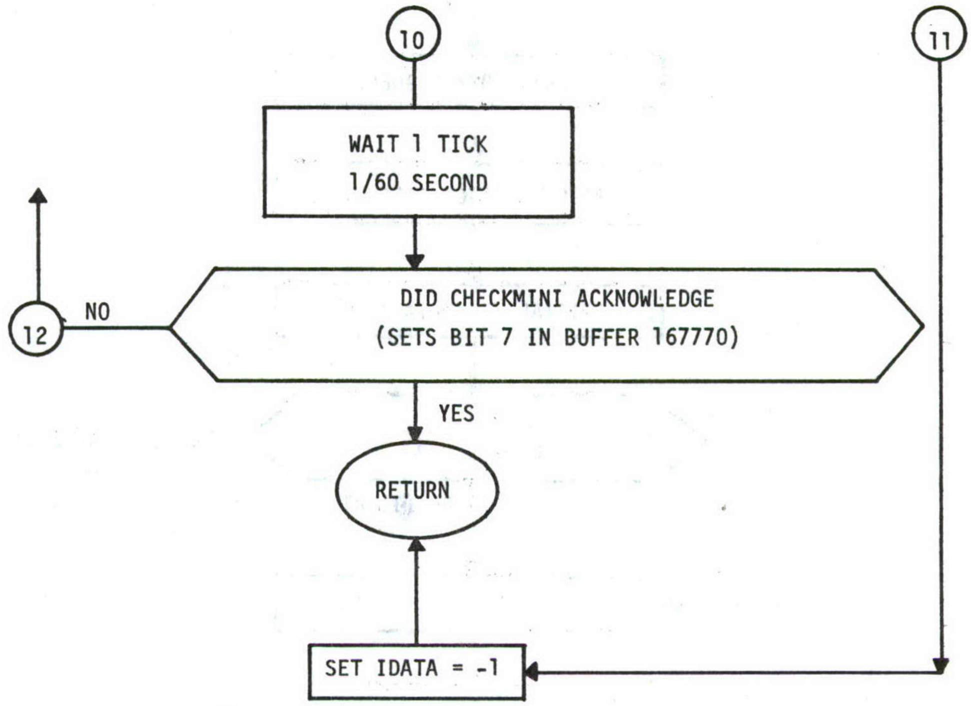
SUBROUTINE IHOLD (ITICK)



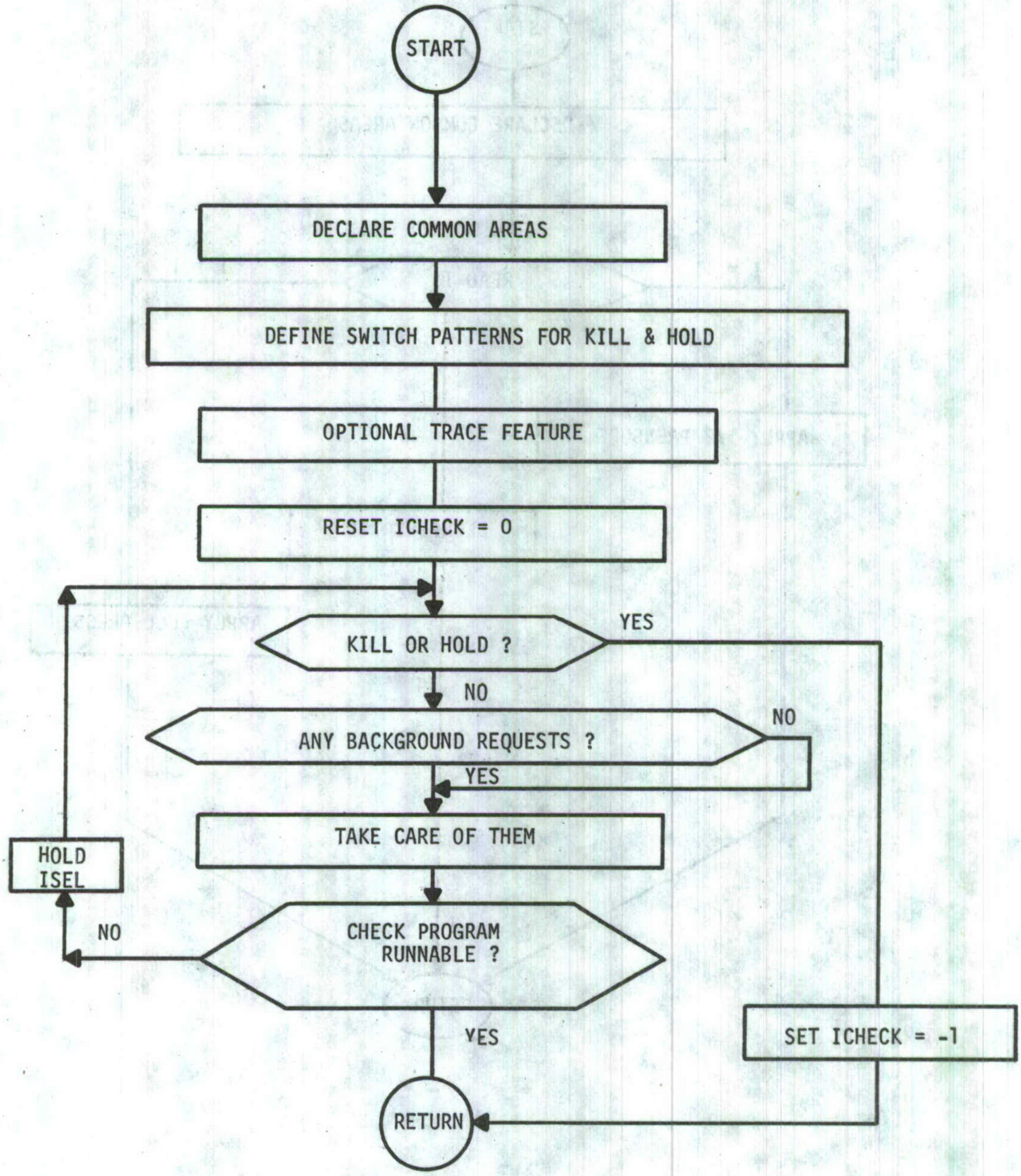
FUNCTION IDATA (LEV, IREFS)



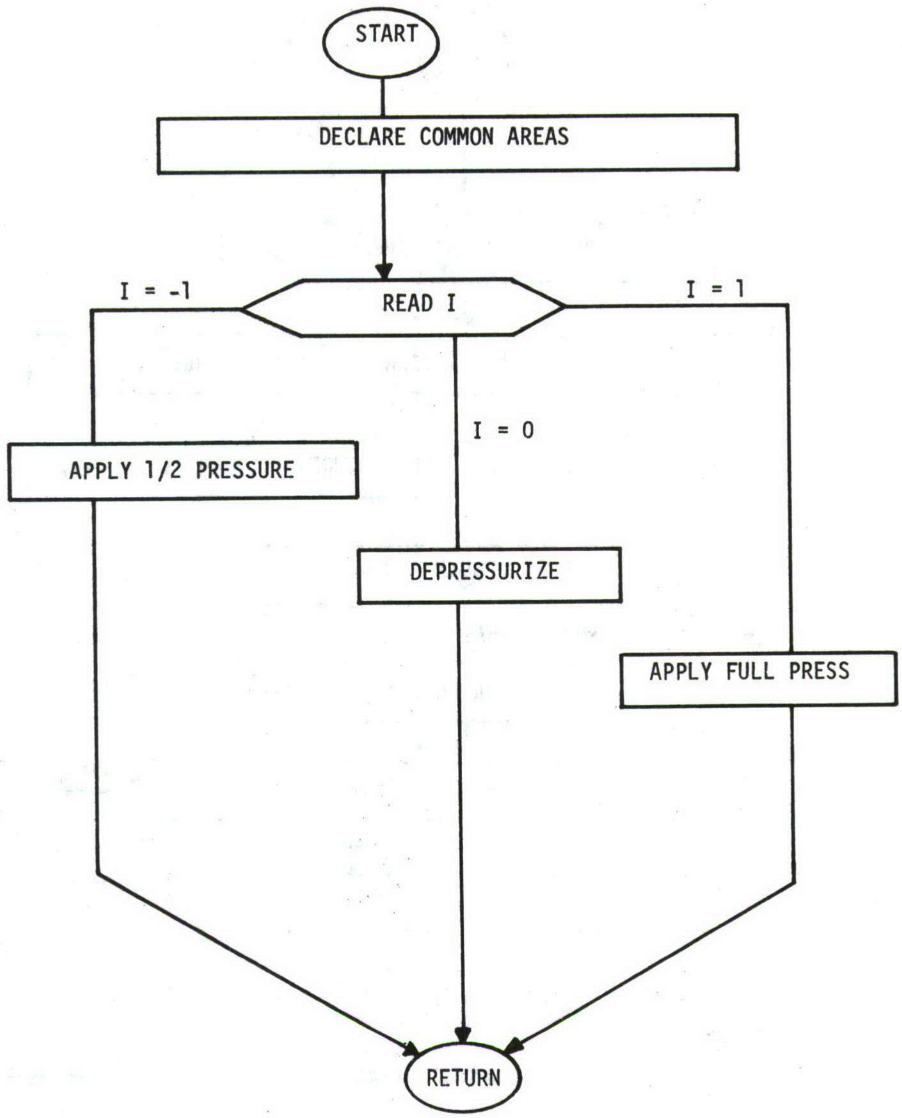
FUNCTION IDATA (CONTINUED)



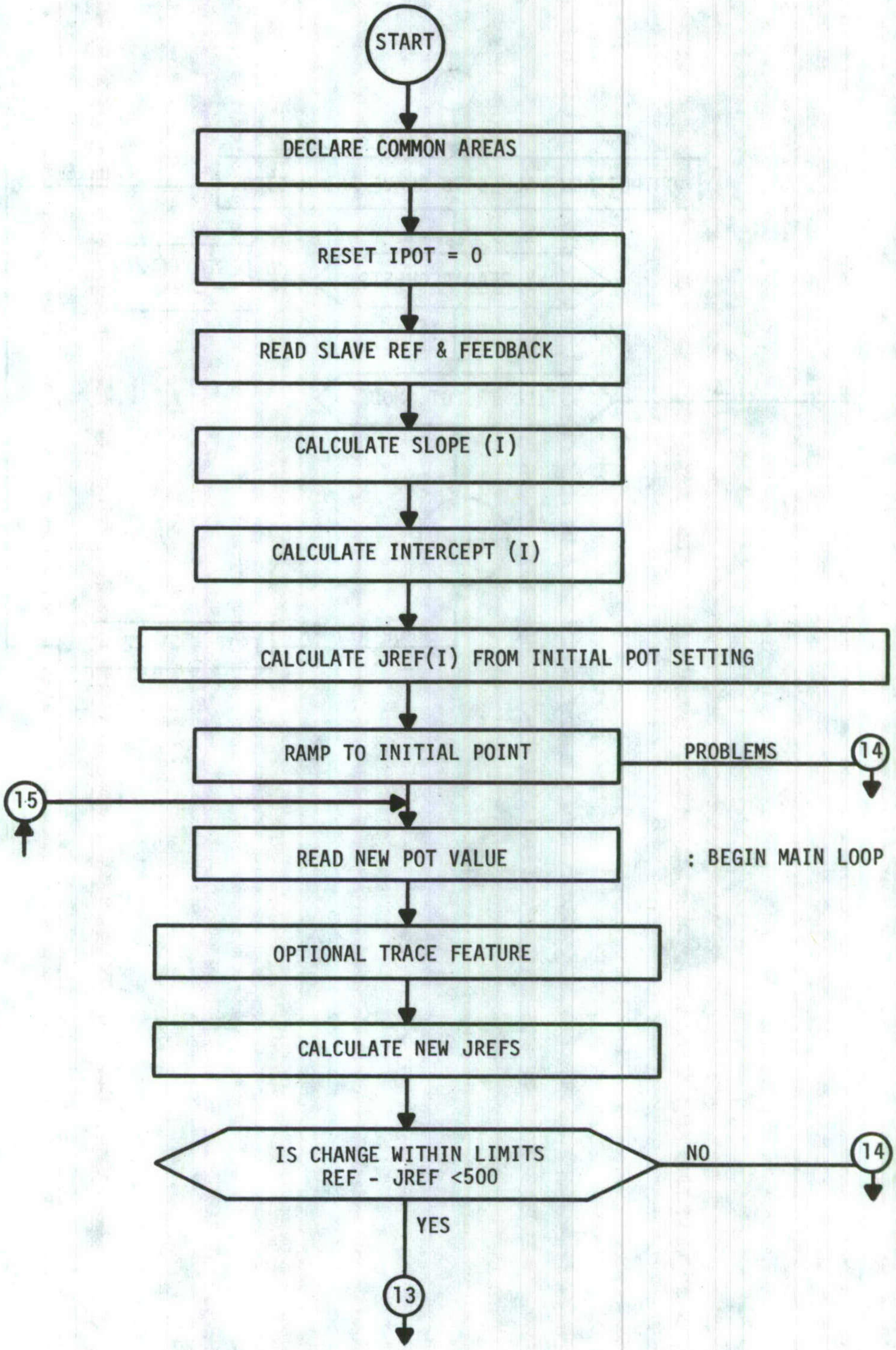
FUNCTION ICHECK



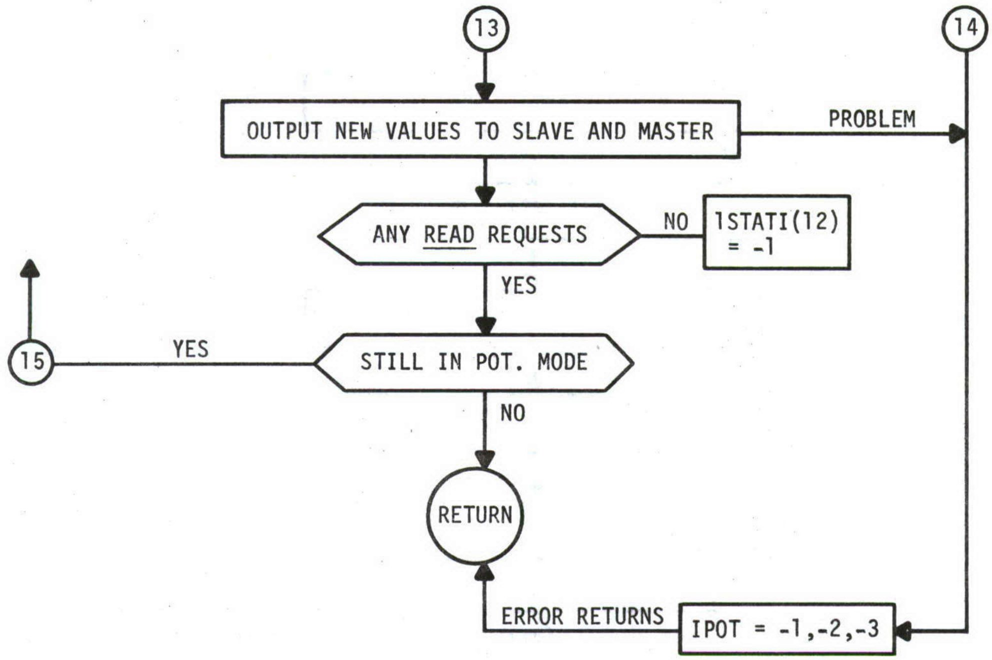
SUBROUTINE IAIR (I)

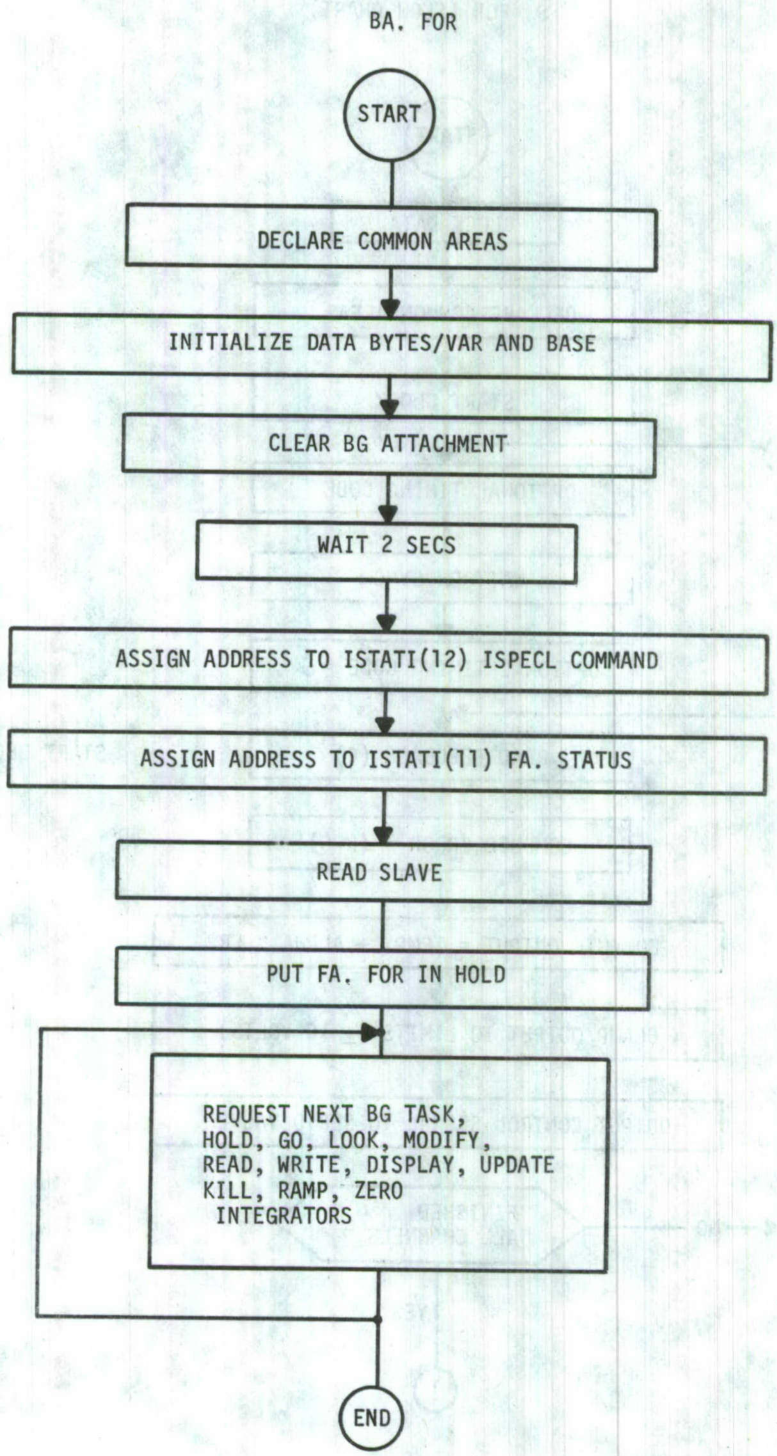


FUNCTION IPOT

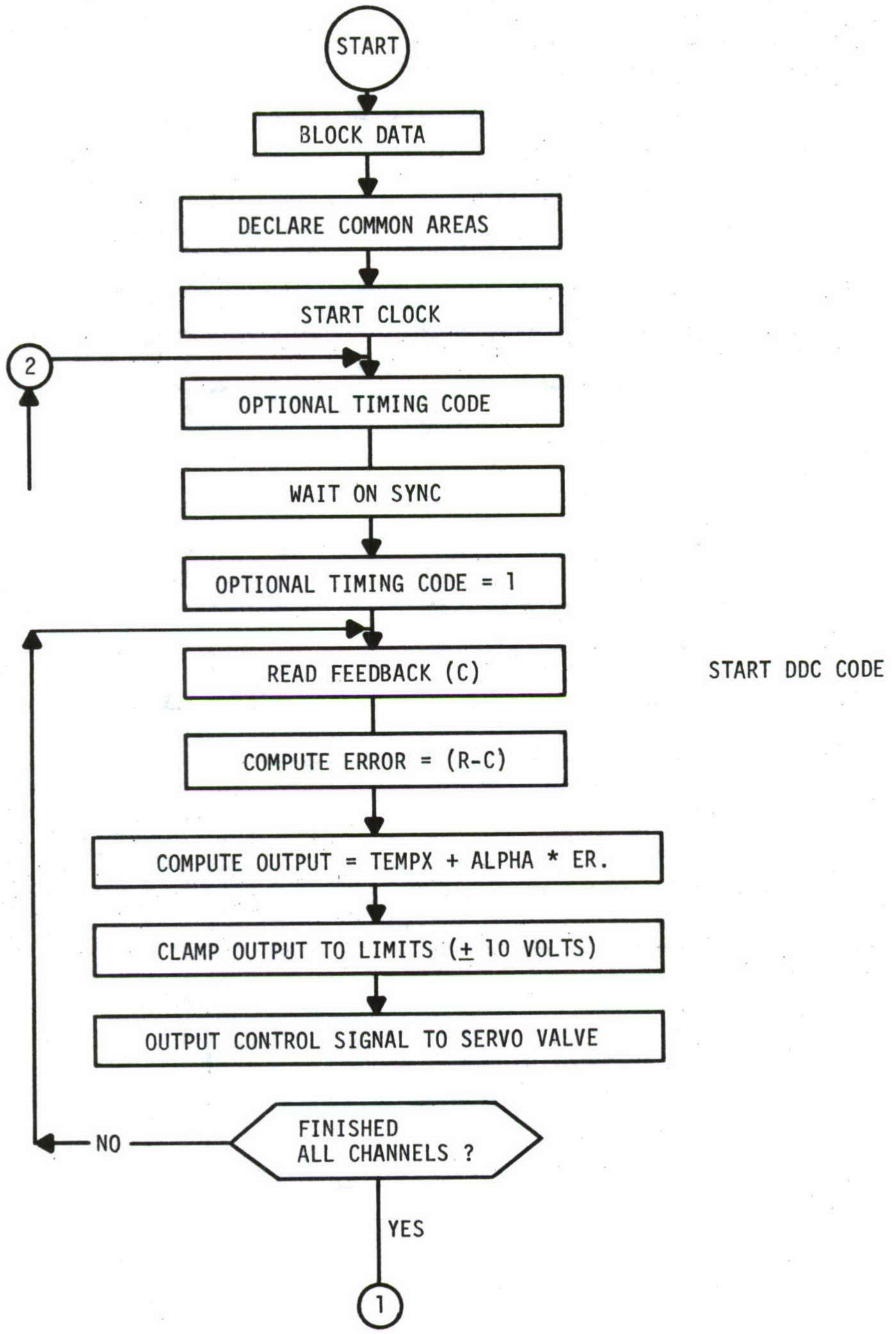


FUNCTION IPOT (CONTINUED)

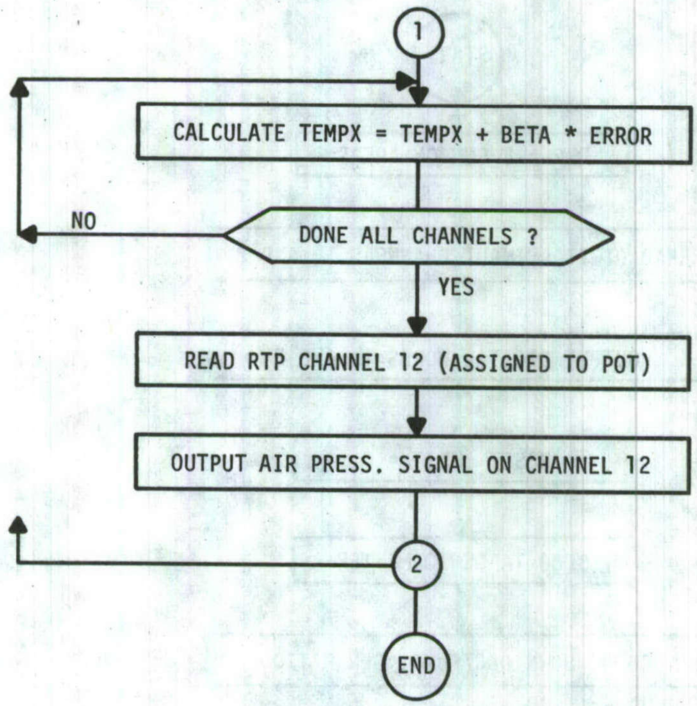



BA. FOR
START
DECLARE COMMON AREAS
INITIALIZE DATA BYTES/VAR AND BASE
CLEAR BG ATTACHMENT
WAIT 2 SECS
ASSIGN ADDRESS TO ISTATI(12) ISPECL COMMAND
ASSIGN ADDRESS TO ISTATI(11) FA. STATUS
READ SLAVE
PUT FA. FOR IN HOLD
REQUEST NEXT BG TASK, HOLD, GO, LOOK, MODIFY, READ, WRITE, DISPLAY, UPDATE KILL, RAMP, ZERO INTEGRATORS
END

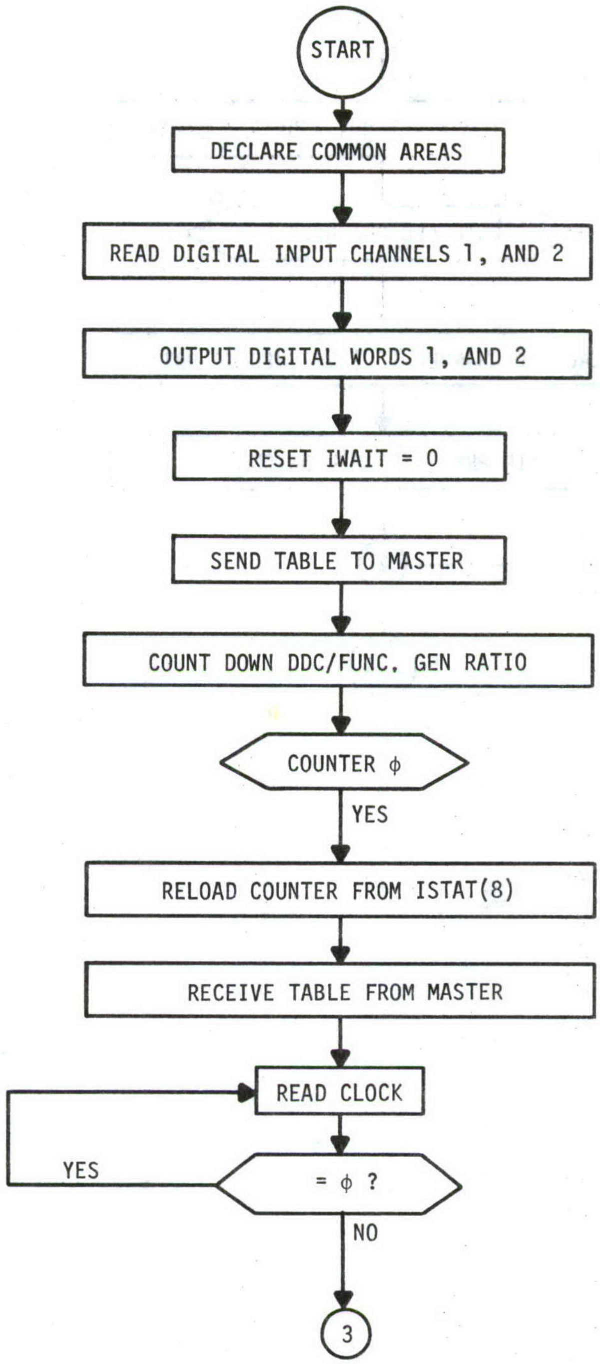
S. FOR (FLOW CHART)



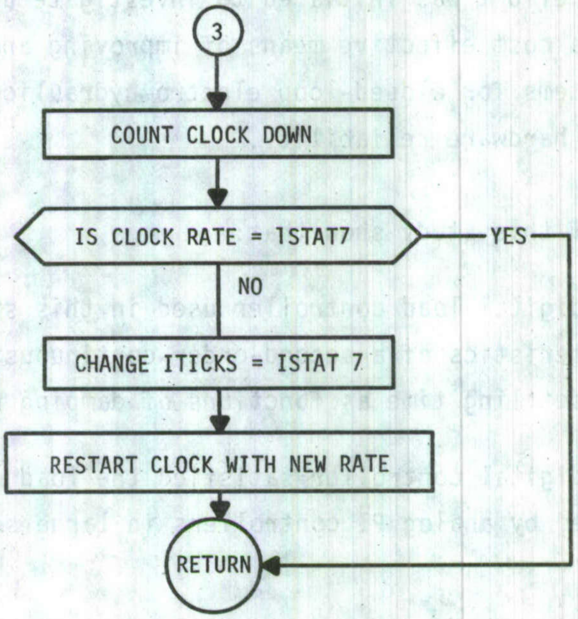
S. FOR (CONTINUED)



FUNCTION IWAIT (PART OF S. FOR)



FUNCTION IWAIT S. FOR (CONTINUED)



SECTION VIII

CONCLUSIONS

This work effort was initiated to investigate use of digital techniques as a cost effective means of improving and simplifying test supporting systems for closed-loop electro-hydraulic load control, with an increase in hardware reliability.

Results of this study show that:

(1) The digital load controller used in this study displays the typical characteristics of a second-order continuous controller with peak overshoot and settling time as functions of damping ratio.

(2) The digital controller satisfied the load control requirements usually provided by analog PI controllers in large scale multi-channel fatigue tests.

(3) The controller algorithm is simple, with two on-line computations per channel (Ref. Section IV and Appendix A). There are no mathematical terms for the effects of adjacent channels, that is, each controller channel is responsible for its own load.

(4) The above assumptions hold true for servo-valve actuators with a frequency response below 100 Hz (the type used in the Structures Test Facility at WPAFB, Ohio).

(5) Structural resonances may be controlled with simple notch filters, as demonstrated on the test stand.

(6) The digital system is inherently capable of manipulating a much greater type and variety of test information than its analog counterpart with flexible formats of presentation on its cathode ray terminal display systems.

(7) Standard programming languages such as FORTRAN are adequate for this type application with a few programmer-defined calls for non-standard peripherals.

(8) A general purpose minicomputer or microprocessor may be used for digital load control, thus reducing the special purpose equipment required for fatigue test support.

(9) Reliability of digital systems can be high. Failures in the DDC machine (slave minicomputer) accounted for a total of 19 hours down-time for the period 1 July 1976 through 1 July 1977. Much higher failure-caused down-time was experienced with associated equipment and peripherals.

SECTION IX

RECOMMENDATIONS

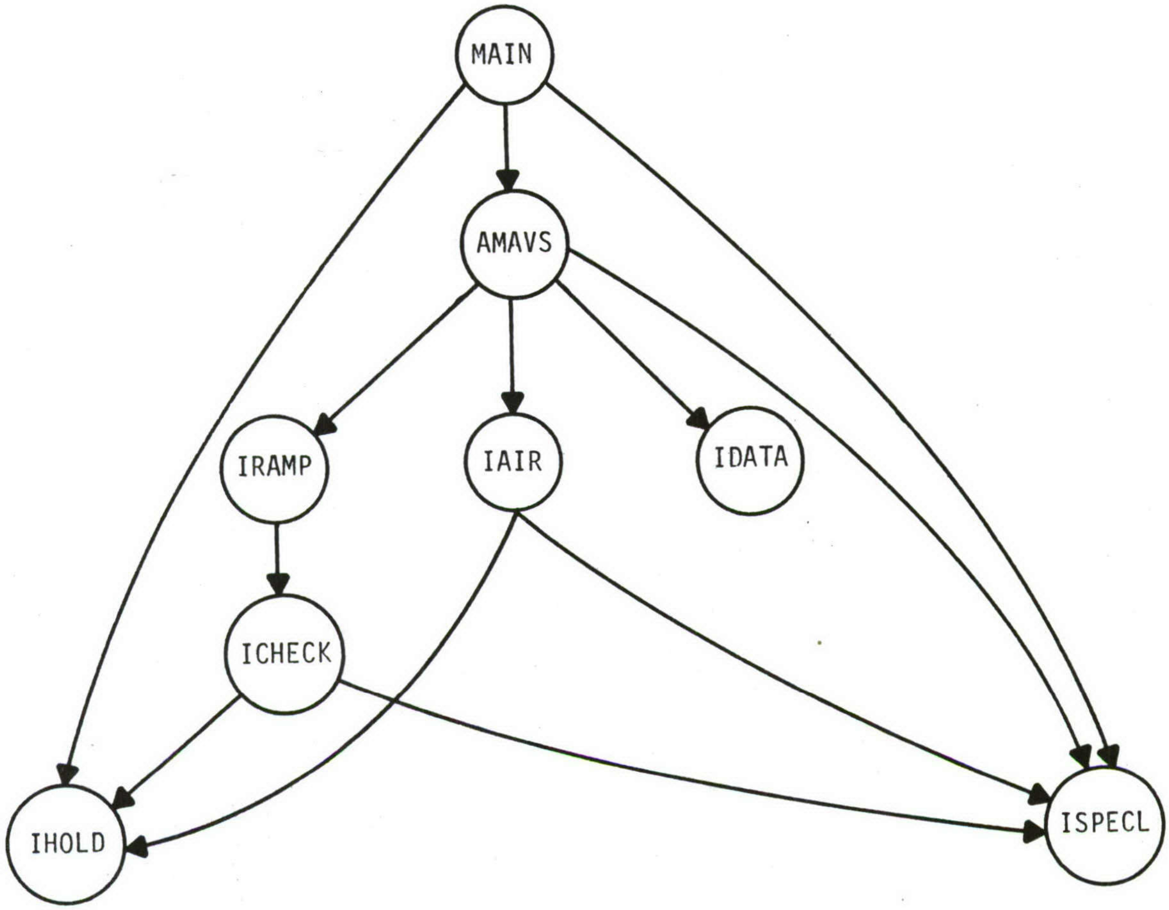
It is recommended that experimental work in digital controls be extended towards (1) refinements in control algorithms, interactive controls and displays and (2) development of large-scale (100-200 channel capacity) systems.

APPENDIX A

COMPUTER PROGRAM LISTINGS

COMPUTER PROGRAM LISTINGS

FA. FOR (ARCHITECTURE)



LISTINGS FOLLOW.

```
C       FA.FOR 13-JAN-77
C       REFLECTS RESCALED BLOCKD.SCALE FACTOR IN AMSGEN=4095
C       FOREGROUND PROGRAM F MODIFIED TO SUPPORT ONLY AMAVS
C       DECLARE COMMON AREAS & LOCAL ARRAYS & INIT LOCAL ARRAYS
C       COMMON DEFINITIONS
        COMMON IN(2),IOUT(2)
        COMMON IR(12),IC(12),IO(12)
        COMMON RE(12),ALPHA(12),BETA(12),TEMPX(12)
        COMMON ISTAT1(12),ISTAT2(12),JREF(12)
        COMMON MIN(12),MAX(12),SLOPE(12),BINTER(12)
        COMMON ICYCLE(281),NPTS(281),IREF(11,281)
        EQUIVALENCE (JRAMP,ISTAT2(12))
C       BEGIN EXECUTABLE CODE SECTION
C
C       INITIALIZATION
        IF(INITRT().NE.0)STOP 'INITRT FAILURE'
        IF(IQSET(5).NE.0)STOP 'QSET FAILURE'
C       LET CONSOLE OPERATOR KNOW FOREGROUND ALIVE
        TYPE 100
100     FORMAT(' FOREGROUND HAS STARTED')
C
C       BEGIN MAINLINE LOOP
C
C       IF ANY SPECIAL REQUESTS FROM BACKGROUND DO THEN NOW
10      IF(ISTAT1(12).NE.0)ISTAT1(12)=ISPECL(ISTAT1(12))
C
C
C       IF SPECIAL POT MODE SELECTED THEN CALL POT MODE
        IF(ISTAT1(11).EQ.3)ISTAT1(11)=IPOT()
C
C       CHECK RAMP CONSTANT THEN IF RAMP REQUESTED DO IT
        IF(JRAMP.LE.100)JRAMP=1000
        IF(ISTAT1(11).EQ.2)ISTAT1(11)=IRAMP(JREF(1),JRAMP,0)
C
C
C       IF A RUN HAS BEEN REQUESTED RUN AMAVS
        IF(ISTAT1(11).EQ.1)CALL AMAVS
C
C
C       OPTIONAL TRACE FEATURE FOR DEBUGGING
D       ISTAT1(9)=ISTAT1(9).OR.2**0
C
C       ALLOW  FOR BACKGROUND REATTACHMENT
        CALL IPOKE("400,IADDR(IN(1)))
C       SLEEP FOR A BIT
        CALL IHOLD(10)
C
C       RETURN FOR NEXT ITTERATION
        GO TO 10
        END
C
        FUNCTION AMAVS
C       COMMON DEFINITIONS
        COMMON IN(2),IOUT(2)
```

```
      COMMON IR(12),IC(12),IO(12)
      COMMON RE(12),ALPHA(12),BETA(12),TEMPX(12)
      COMMON ISTAT1(12),ISTAT2(12),JREF(12)
      COMMON MIN(12),MAX(12),SLOPE(12),BINTER(12)
      COMMON ICYCLE(281),NPTS(281),IREF(11,281)
      DIMENSION IZERO(12)
      DATA IZERO/12*0/
C     EQUIVALENCE FOR EASE OF UNDERSTANDING CODE
      EQUIVALENCE (IMIS,ISTAT2(1)),(ILEV,ISTAT2(2)),(ICYC,ISTAT2(3))
      EQUIVALENCE (NEWMIS,ISTAT2(4)),(NEWLEV,ISTAT2(5)),(NEWCYC,ISTAT2
(6))
      EQUIVALENCE (NEWMAX,ISTAT2(7)),(KRAMP,ISTAT2(8))
      EQUIVALENCE (KHOLD,ISTAT2(9)),(KBREAK,ISTAT2(10)),(KNOLIN,ISTAT2
(11))
      EQUIVALENCE (JRAMP,ISTAT2(12))
C
C     BEGIN EXECUTABLE CODE FOR AMAVS FUNCTION GENERATION
C
C     OPTIONAL TRACE FEATURE FOR DEBUGGING
D     ISTAT1(9)=ISTAT1(9).OR.2**1
C
C     INITIALIZATION
C     NKM-6  RESET DATA BUFFER
C     INSURE 0<KRAMP<=100
      IF(KRAMP.LE.0.OR.KRAMP.GT.100)KRAMP=6
C     INSURE KHOLD >= 60 TICKS
      IF(KHOLD.LT.60)KHOLD=100
C     INSURE NEWMIS<=NEWMAX<=1280 & >0
      IF(NEWMAX.LT.NEWMIS.OR.NEWMAX.GT.1280.OR.NEWMAX.LE.0)NEWMAX=1280
C     INSURE INITIAL MISSION WITHIN RANGE
      IMIS0=NEWMIS
      IF(IMIS0.LE.0.OR.IMIS0.GT.1280)IMIS0=1
C     CHECK THAT 0<BREAK POINT<16000
      IF(KBREAK.LE.0.OR.KBREAK.GT.16000)KBREAK=3000
C     CHECK THAT 1<=NONLINEAR MULTIPLIER FOR RAMP ABOVE BREAK
      IF(KNOLIN.LT.1)KNOLIN=2
C     INITIAL LAST POINT = FIRST POINT
      IOLD=1
C     LET BRIDGE KNOW FOREGROUND ALIVE & READY
      CALL IAIR(-1)
C     LET DATA KNOW WE ARE READY TO START
      IF(IDATA(0).NE.0)GO TO 999
C
C
C     FUNCTION GENERATION CODE FOLLOWS
      DO 300 IMIS=IMIS0,NEWMAX
C     CALL AIR PRESSURIZATION
      CALL IAIR(1)
      DO 200 ILEV=1,281
      N=ICYCLE(ILEV)
      IF(N.GT.0)GOTO 30
      IF(N.EQ.-1.AND.IMIS/10*10.NE.IMIS)GOTO 200
      IF(N.EQ.-2.AND.IMIS/100*100.NE.IMIS)GOTO 200
```

```
      N=1
C
30    DO 100 ICYC=1,N
C     CALCULATE NUMBER OF POINTS IN RAMP & MAKE SURE NON ZERO
      ISTEPS=NPTS(ILEV)/KRAMP
      IF(ISTEPS.LE.0)ISTEPS=1
C     HANDLE THE NONLINEAR CASE
      IF(NPTS(ILEV).GE.KBREAK)ISTEPS=ISTEPS*KNOLIN
C
C     ALLOW FOR ANY SPECIAL REQUESTS FROM BACKGROUND
C     DETECT KILL REQUESTS & HANDLE HOLD & SPECIAL REQUESTS
C     DH1:27-JAN-77 CHANGED TO HANDLE SIMULTANEOUS GO AND ABORT
      IF(ICHECK().GE.0)GOTO 35
      IF(NEWMIS.NE.0)GO TO 999
      GO TO 900
35    CONTINUE
C
C     IF NO REQUEST FOR SPECIAL START POINT SKIP CODE
      IF(NEWMIS.EQ.0)GOTO 40
      IF(NEWMIS.NE.IMIS.OR.NEWLEV.NE.ILEV.OR.NEWCYC.NE.ICYC)GOTO 100
      NEWMIS=0
      NEWLEV=0
      NEWCYC=0
      ISTEPS=JRAMP
C
C     NKM-5  DATA REQUIRES A ZERO AT END OF FLITE
40    LEVEL=ILEV
      IF(LEVEL.EQ.281)LEVEL=0
      IF(LEVEL.EQ.257.AND.IMIS/10*10.NE.IMIS)LEVEL=0
C     IF ODD RAMP TO NEW POINT - IF EVEN RETURN TO PREVIOUS POINT
      IF(ICYC.EQ.ICYC/2*2)GOTO 50
C     NKM-2 CALL DATA ONLY THE FIRST CYCLE
C     CALL DATA THEN RAMP THEN HOLD
      IF(ICYC.GT.1)GO TO 45
      IF(IDATA(LEVEL).NE.0)GOTO 900
45    IF(IRAMP(IREF(1,ILEV),ISTEPS,1).NE.0)GOTO 900
C     NKM-2 BIT 14 SET TO INDICATE RAMP TOP
      LEVTOP=LEVEL+"40000
      IF(IDATA(LEVTOP).NE.0)GO TO 900
      CALL IHOLD(KHOLD)
      GOTO 100
C
C     CALL  ,RAMP & HOLD (GOTO PREVIOUS LEVEL)
C     NKM-2 REMOVED CALL TO DATA
50    IF(IRAMP(IREF(1,IOLD),ISTEPS,1).NE.0)GOTO 900
      CALL IHOLD(KHOLD)
C
100   CONTINUE
      IOLD=ILEV
200   CONTINUE
C     NKM-4 DEPRESSURIZE EVERY MISSION COMPLETION
      CALL IAIR(0)
300   CONTINUE
C
C     ABORT RETURN/SET UP FOR RESTARTABILITY
900   NEWMIS=IMIS
      NEWLEV=ILEV
      NEWCYC=ICYC
```

```
          ISTAT1(11)=0
          ISTAT1(12)=0
          DO 910 I=1,12
910       TEMPX(I)=0.
C         NOW SAVE INFO ON DISK FOR RESTART
          IDUM=ISPECL(3)
C         RAMP TO ZERO VERY SLOWLY
          CALL IRAMP(IZERO,1000,0)
C         NKM-4LEAVE AIR PRESSURE ON
          CALL IPOKE("167772,0)
999       RETURN
          END
C
C
          FUNCTION ISPECL(IVAR)
C         HANDLE ALL FORGROUND I/O REQUESTS TO DISK & SLAVE
C         INCLUDING SPECIAL REQUESTS OF THE BACKGROUND HERE
C         SPECIFICALLY
C         IVAR=1 TRANSMIT ARRAY TO SLAVE
C         IVAR=2 RECEIVE ARRAY FORM SLAVE
C         IVAR=3 WRITE OUT COMMON TO DISK
C         IVAR=4 READ IN OLD COMMON FROM DISK
C         IVAR=5 TRANSMIT NEW IN,IOUT,IR TO SLAVE
C         IVAR=6 RECEIVE FROM SLAVE CURRENT IN,IOUT,IR
C
C         COMMON DEFINITIONS
          COMMON IN(2),IOUT(2)
          COMMON IR(12),IC(12),IO(12)
          COMMON RE(12),ALPHA(12),BETA(12),TEMPX(12)
          COMMON ISTAT1(12),ISTAT2(12),JREF(12)
          COMMON MIN(12),MAX(12),SLOPE(12),BINTER(12)
          COMMON ICYCLE(281),NPTS(281),IREF(11,281)
C
C         OPTIONAL TRACE FEATURE FOR DEBUGGING
D         ISTAT1(9)=ISTAT1(9).OR.2**2
C
C
C         ALLOW FOR BACKGROUND (RE)ATTACHMENT
50        CALL IPOKE("400,IADDR(IN(1)))
C
          IF(IVAR.EQ.0)RETURN
          GOTO(1,2,3,4,5,6),IVAR
          GOTO 60
C
C
C         SEND TO SLAVE
1         IF(MSND(IN(1),144).NE.0)STOP 'FATAL SEND TO SLAVE'
          GOTO 60
C
C         RECEIVE FROM SLAVE
2         IF(MREC(IN(1),144).NE.0)STOP 'FATAL RECEIVE FROM SLAVE'
          GOTO 60
C
C         WRITE COMMON AREAS TO DISK
3         CALL ASSIGN(1,'SY:AMACOM.DAT')
```

```
      WRITE(1)ALPHA,BETA,TEMPX,ISTAT1,ISTAT2,JREF
      CALL CLOSE(1)
      GOTO 60
C
C     READ COMMON AREAS FROM DISK
4     CALL ASSIGN(1,'SY:AMACOM.DAT')
      READ(1)ALPHA,BETA,TEMPX,ISTAT1,ISTAT2,JREF
      CALL CLOSE(1)
      GOTO 60
C
C
C
C     SEND IN,IOUT,IR TO SLAVE
5     IF(MSND(IN(1),16).NE.0)STOP 'FATAL SEND TO SLAVE (16)'
      GOTO 60
C
C     RECEIVE IN,IOUT,IR FROM SLAVE
6     IF(MREC(IN(1),16).NE.0)STOP 'FATAL RECEIVE FROM SLAVE(16)'
      GOTO 60
C     END OF SPECIAL FUNCTION ROUTINES
C     ALWAYS RETURNS ZERO (MAKES BACKGROUND REQUESTS CLEANER)
60    ISPECL=0
C
C
C
      RETURN
      END
C
C
      FUNCTION IRAMP(NEW,NUM,IABORT)
C     DECLARE COMMON AREAS & LOCAL ARRAYS & INIT LOCAL ARRAYS
C     COMMON DEFINITIONS
      COMMON IN(2),IOUT(2)
      COMMON IR(12),IC(12),IO(12)
      COMMON RE(12),ALPHA(12),BETA(12),TEMPX(12)
      COMMON ISTAT1(12),ISTAT2(12),JREF(12)
      COMMON MIN(12),MAX(12),SLOPE(12),BINTER(12)
      COMMON ICYCLE(281),NPTS(281),IREF(11,281)
      DIMENSION NEW(12),DELTA(12),REF(12)
C     NKM-1 ADDED IRM(11) FOR MASTER ANALOGUE OUTPUTS
      DIMENSION IRM(11)
C
C     BEGIN EXECUTABLE CODE
C     OPTIONAL TRACE FEATURE FOR DEBUGGING
D     ISTAT1(9)=ISTAT1(9).OR.2**3
C
C     READ IR FROM SLAVE
      IDUM=ISPECL(6)
C
C     LIM CHECK,CALCULATE DELTA'S & ESTABLISH INITIAL FLOATING REF
      DO 20 I=1,11
      IF(IABS(NEW(I)).GE.8192)GOTO 999
      TMP=NEW(I)-IR(I)
      DELTA(I)=TMP/NUM
2
```

```
 0      REF(I)=IR(I)
C
C       NOW GO UP/DOWN THE RAMP
        DO 50 J=1,NUM
        DO 40 I=1,11
        REF(I)=REF(I)+DELTA(I)
C       NKM-1 CALCULATE MASTER OUTPUT SCALED TO 4095
        IRM(I)=REF(I)*.5
40      IR(I)=REF(I)
C       NKM-1 NOW CHECK FOR ANALOG OUT INHIBIT
C       & SKIP IF INHIBITED
        IF(ISTAT1(10).AND.2)GOTO 45
C       WRITE OUT VALUES TO DATA ON ANALOGUE CHANNELS
        IF(ID2A(IRM).NE.0)GOTO 998
C       ERROR REPORT SAME
C       OUTPUT NEW IR TO SLAVE
45      IDUM=ISPECL(5)
C       EVERY 8TH ITTERATION SEE ABOUT REQUESTS FROM
C       THE BRIDGE (BUTTONS OR CONSOLE)
        IF((J.AND.7).OR.(IABORT.EQ.0))GOTO 50
        IF(ICHECK().NE.0)GOTO 999
50      CONTINUE

C
C       NORMAL RETURN
        IRAMP=0
        RETURN
C
C       ERROR RETURN
998     IRAMP=-2
        RETURN
999     IRAMP=-1
        RETURN
        END
C
C
        SUBROUTINE IHOLD(ITICK)
        CALL ISLEEP(0,0,0,ITICK)
        RETURN
        END
C
C
        FUNCTION IDATA(LEV,IREFS)
C       COMMON DEFINITIONS
        COMMON IN(2),IOUT(2)
        COMMON IR(12),IC(12),IO(12)
        COMMON RE(12),ALPHA(12),BETA(12),TEMPX(12)
        COMMON ISTAT1(12),ISTAT2(12),JREF(12)
        COMMON MIN(12),MAX(12),SLOPE(12),BINTER(12)
        COMMON ICYCLE(281),NPTS(281),IREF(11,281)
C       ASSUME SUCCESSFUL
C       OPTIONAL TRACE FEATURE FOR DEBUGGING
D       ISTAT1(9)=ISTAT1(9).OR.2**4
C
        IDATA=0
C
C       IF INHIBIT DATA SET SKIP OVER DATA CODE
```

```
      IF(ISTAT1(10).AND.1)GOTO 90
C
C     HANDLE COMMUNICATIONS WITH DATA CHECK MINI
C     OUTPUT LEVEL PLUS FLAG
      CALL IPOKE("167772,"100000.OR.LEV)
C     CHECK FOR KILL
10    IF(ISTAT1(11).LT.0)GOTO 998
C     WAIT A TICK
      CALL IHOLD(1)
C     IF DATA NORESPONSIVE TRY AGAIN
      IF(.NOT.IPEEK("167770).AND.2**7)GOTO 10
C     RESPOND BY ZEROING OUT BUFFER
      CALL IPOKE("167772,0)

C     CHECK FOR KILL
20    IF(ISTAT1(11).LT.0)GOTO 998
C     WAIT FOR A TICK
      CALL IHOLD(1)
      IF(IPEEK("167770).AND.2**7)GOTO 20
C
C     NKM-1
C     REMOVED MASTER ANALOGUE OUTPUT TO IRAMP FUNCTION
C     NKM-1
90    RETURN
C
C     ERROR RETURNS
998   IDATA=-1
      RETURN
      END
C
C
      FUNCTION ICHECK
C     PURPOSE IS TO CHECK FOR
C     KILL, STOP, GO REQUESTS FROM CONSOLE OR BUTTONS
C     AND READ REQUESTS FROM CONSOLE (BACKGROUND)
C     COMMON DEFINITIONS
      COMMON IN(2),IOUT(2)
      COMMON IR(12),IC(12),IO(12)
      COMMON RE(12),ALPHA(12),BETA(12),TEMPX(12)
      COMMON ISTAT1(12),ISTAT2(12),JREF(12)
      COMMON MIN(12),MAX(12),SLOPE(12),BINTER(12)
      COMMON ICYCLE(281),NPTS(281),IREF(11,281)
C
C     DATA DEFINITION OF BIT PATTERNS FOR SWITCHES
      DATA ISKILL/"40000/,ISHOLD/"10000/
C
C     BEGIN EXECUTABLE CODE
C
C     ASSUME SUCCESSFUL
C     OPTIONAL TRACE FEATURE FOR DEBUGGING
D     ISTAT1(9)=ISTAT1(9).OR.2**5
C
      ICHECK=0
C
C     READ THE KILL,HOLD, SWITCHES FROM BRIDGE
1
```

```
 0      CALL ISPECL(6)
C
C       CHECK FOR KILL (ABORT)
        IF(ISTAT1(11).LT.0.OR.IN(1).AND.ISKILL)GOTO 998
C
C       CHECK FOR SPECIAL REQUESTS FROM BACKGROUND
        IF(ISTAT1(12).NE.0)ISTAT1(12)=ISPECL(ISTAT1(12))
C
C       IF WE ARE RUNABLE THEN RETURN
        IF(ISTAT1(11).GE.1.AND.(.NOT.IN(1).AND.ISHOLD))RETURN
C
C       WAIT JUST ONE SECOND
        CALL IHOLD(60)
C       GO BACK FOR ANOTHER CHECK
        GOTO 10
C
C       KILL (ABORT) RETURN POINT
998     ICHECK=-1
        RETURN
        END
C
C
        SUBROUTINE IAIR(I)
C       COMMON DEFINITIONS
        COMMON IN(2),IOUT(2)
        COMMON IR(12),IC(12),IO(12)
        COMMON RE(12),ALPHA(12),BETA(12),TEMPX(12)
        COMMON ISTAT1(12),ISTAT2(12),JREF(12)
        COMMON MIN(12),MAX(12),SLOPE(12),BINTER(12)
        COMMON ICYCLE(281),NPTS(281),IREF(11,281)
C       OPTIONAL TRACE FEATURE FOR DEBUGGING
D       ISTAT1(9)=ISTAT1(9).OR.2**6
C
        IF(I)5,10,20
C
C       I < 0 THEN SIGNAL OPERATIONAL
5       CALL ISPECL(6)
        IR(12)=2000
        CALL ISPECL(5)
        RETURN
C
C       I=0 DEPRESURIZE
10      CALL ISPECL(6)
        IR(12)=0
        CALL ISPECL(5)
        CALL IHOLD(480)
        RETURN
C
C
C       I > 0 THEN PRESURIZE
20      CALL ISPECL(6)
        IR(12)=6000
        CALL ISPECL(5)
        CALL IHOLD(900)
```

```
      RETURN
      END
C
C
      FUNCTION IPOT
C     PURPOSE OF THIS FUNCTION IS TO UTILIZE THE POTENTIOMETER
C     ON CHANNEL 12 FOR ALLOWING MANUAL CHANGING OF THE
C     LOADS TO THE CHANNELS
C     ARRAYS USED IR,JREF,MIN,MAX,SLOPE,BINTER
C
C     COMMON DEFINITIONS
      COMMON IN(2),IOUT(2)
      COMMON IR(12),IC(12),IO(12)
      COMMON RE(12),ALPHA(12),BETA(12),TEMPX(12)
      COMMON ISTAT1(12),ISTAT2(12),JREF(12)
      COMMON MIN(12),MAX(12),SLOPE(12),BINTER(12)
      COMMON ICYCLE(281),NPTS(281),IREF(11,281)
      DIMENSION IRM(11)
C
C     BEGIN EXECUTABLE CODE SETION
C
C     INITIALIZATION
C     ASSUME SUCCESSFUL
      IPOT=0
C     READ IR & IC FROM SLAVE
      CALL ISPECL(2)
C     SET UP SLOPES AND INTERCEPTS AND INITIAL POINT
      DO 10 I=1,11
      SLOPE(I)=FLOAT(MAX(I)-MIN(I))/4096.
      BINTER(I)=MIN(I)
10    JREF(I)=SLOPE(I)*IC(12)+BINTER(I)
C
C     RAMP TO INITIAL POINT
      IF(IRAMP(JREF,2000,1).NE.0)GOTO 998
C
C     BEGIN MAIN LOOP
C     READ NEW POT VALUE
30    CALL ISPECL(2)
C
C     OPTIONAL TRACE FEATURE FOR DEBUGGING
D     ISTAT1(9)=ISTAT1(9).OR.2**7
C
C     CALCULATE NEW JREFS & INSURE IN BOUNDS
      DO 20 I=1,11
      JREF(I)=SLOPE(I)*IC(12)+BINTER(I)
      IF(IABS(IR(I)-JREF(I)).GT.500)GOTO 997
      IR(I)=JREF(I)
      IRM(I)=IR(I)/2
20    CONTINUE
C
C     OUTPUT THE NEW VALUES
      IF(ID2A(IRM).NE.0)GO TO 996
      CALL ISPECL(5)
C
C     HANDLE READ SPECIAL REQUEST
```

```
      IF(ISTAT1(12).EQ.2)ISTAT1(12)=ISPECL(ISTAT1(12))
C     ALL OTHERS ARE ERRORS
      IF(ISTAT1(12).NE.0)ISTAT1(12)=-1
C
C     IF POT MODE STILL IN EFFECT GO ONCE AGAIN
      IF(ISTAT1(11).EQ.3) GOTO 30
C
C     OTHERWISE RETURN
      RETURN
C
C     ERROR RETURNS
996   IPOT=IPOT-1
997   IPOT=IPOT-1
998   IPOT=IPOT-1
      RETURN
      END
*
```

```
C       BA.FOR 21-JAN-77 MODIFICATION OF B .FOR
C       FOR USE WITH AMAYS
C       ALL REFERENCES TO SNOOPY DEMO
C       HAVE BEEN REMOVED
C
C       UTILIZING PEEKS & POKES
C
        BLOCK DATA
        COMMON L,LF,LR,MBPY(11),MBASE(11)
C
        DATA MBPY/2,2,2,2,4,4,4,4,2,2,2/
        DATA MBASE/0,8,32,56,80,128,176,224,272,296,320/
        END
C
        COMMON L,LF,LR,MBPY(11),MBASE(11)
        DIMENSION SF(4),SV(4)
        DATA SF/'HGLM','RWDU','K@PZ',0./
        DATA SV/'URCO','EABT','12@',0./
C
C
C       MAINLINE
        CALL IPOKE("400,0)
        CALL ISLEEP(0,0,2,0)
        L=IPEEK("400)
        IF(L.EQ.0)STOP 'FOREGROUND NOT ACTIVE'
        LF=L+144*2-2+8
        LR=L+143*2-2+8
        CALL IPOKE(LF,2)
        CALL IPOKE(LR,0)
        TYPE 99,L,LF,LR
99      FORMAT(' RUNNING WITH L LF LR ',3O8)
C       MAKE THE SCREEN APPEAR NICE
        GO TO 7
C
C       FUNCTION DISPATCHER
10      CALL ICUR(3)
        TYPE 100
100     FORMAT('+FUNCTION ',$)
        CALL GETSTR(5,SC,1)
        M=INDEX(SF,SC)
        IF(M.EQ.0)GOTO 10
        GOTO(1,2,3,4,5,6,7,8,9,1010,1111,1212),M
C
        GO TO 10
C       STOP
1       CALL IPOKE(LR,0)
        GOTO 10
C
C       RUN
2       CALL IPOKE(LR,1)
        GOTO 10
C
C       LOOK AT A VARIABLE
3       CALL ICUR(3)
        TYPE 110
110     FORMAT('+VARIABLE ',$)
        CALL GETSTR(5,S2,1)
```

```
          M1=INDEX(SV,S2)
          IF(M1.EQ.0)GOTO 10
          CALL ICUR(2)
          TYPE 120
120       FORMAT('+INTEGER CHANNEL# ',$)
          ACCEPT 121,ICHN
121       FORMAT(I3)
          IF(ICHN.LE.0.OR.ICHN.GT.204)GOTO 10
          R=RGVAR(M1,ICHN)
          CALL ICUR(1)
          TYPE 130,S2,ICHN,R
130       FORMAT('+    ',A1,' (',I3,')= ',F15.6)
          CALL ICUR(2)
          GOTO 10
C
C         MODIFY VARIABLE
4         CALL ICUR(1)
          TYPE 130,S2,ICHN,RGVAR(M1,ICHN)
          CALL ICUR(2)
          TYPE 150
150       FORMAT('+ FLOATING POINT NEW VAL= ',$)
          ACCEPT 151,REAL
151       FORMAT(F15.7)
          CALL ICUR(3)
          TYPE 160
160       FORMAT('+ARE YOU SURE? ',$)
          CALL GETSTR(5,SC,3)
          CALL SCOMP('Y',SC,IYES)
          IF(IYES.EQ.0)CALL RPVAR(M1,ICHN,REAL)
          CALL ICUR(2)
          CALL ICUR(1)
          GOTO 10
C
C         READ TABLE FROM SLAVE
5         CALL IPOKE(LF,2)
55        IF(IPEEK(LF).NE.0)GOTO 55
          GOTO 10
C
C         WRITE TABLE TO SLAVE
6         CALL IPOKE(LF,1)
66        IF(IPEEK(LF).NE.0)GOTO 66
          GOTO 10
C
C         DISPLAY
7         CALL ITTOUT("213,"214,1,1)
          TYPE 175
175       FORMAT('+CHAN   REF CONTROL  OUTPUT   ERROR
          9'A        B   TEMPX STAT-1  STAT-2  @JREF')
          TYPE 180,(J,(INT(RGVAR(I,J)),I=2,5),
          9(RGVAR(I,J),I=6,8),
          9(INT(RGVAR(I,J)),I=9,11),J=1,12)
180       FORMAT(' ',I2,4I8,F6.2,F8.4,F8.1,I7,I7,I7)
          TYPE 190,(IPEEK(I),I=L,L+6,2)
190       FORMAT(' ',31X,'IN= ',O7,O7,' OUT= ',O7,O7)
```

```
      GOTO 10
C
C     UPDATE DISK
8     CALL ICUR(3)
      TYPE 700
700   FORMAT('+DISK READ OR WRITE ? ',$)
      CALL GETSTR(5,SC,3)
      M=INDEX(SF,SC)
      IF(M.LT.5.OR.M.GT.6)GOTO 10
      IF(M.EQ.5)CALL IPOKE(LF,4)
      IF(M.EQ.6)CALL IPOKE(LF,3)
77    IF(IPEEK(LF).NE.0)GOTO 77
      GOTO 10
C
C
C     KILL (ABORT)
9     CALL IPOKE(LR,-1)
      GOTO 10
C
C
C     @ (REQUEST RAMP TO VALUES IN JREFARRAY)
1010  CALL IPOKE(LR,2)
      GOTO 10
C
C     DLH-2 NEXT TWO FUNCTIONS ADDED 27-JAN-77
C     POT MODE AND ZERO BETA - TEMPX

C
C
C     ENTER POT MODE
1111  CALL IPOKE(LR,3)
      GOTO 10
C
C
C     ZERO BETA'S & TEMPX
C     !!!DOES NOT WRITE TO SLAVE
C     !!! & ASSUMES POSITION OF BETA & TEMP X IN SV=7 & 8
1212  DO 1213 I=1,12
      CALL RPVAR(7,I,0.)
      CALL RPVAR(8,I,0.)
1213  CONTINUE
      GOTO 10
      END
C
C
      FUNCTION  IGLOC(N,ISUB)
      COMMON L,LF,LR,MBPV(11),MBASE(11)
      IGLOC=L+MBASE(N)+MBPV(N)*(ISUB-1)
      RETURN
      END
C
C
      FUNCTION RGVAR(N,ISUB)
      COMMON L,LF,LR,MBPV(11),MBASE(11)
```

```
      DIMENSION IVAR(2)
      EQUIVALENCE (RVAR,IVAR)
      LOC=IGLOC(N,ISUB)
D     TYPE 100,N,ISUB,LOC,L
D100  FORMAT(' N,ISUB,LOC,L ',2I6,2O7)
      IVAR(1)=IPEEK(LOC)
      IVAR(2)=IPEEK(LOC+2)
      IF(MBPV(N).EQ.2)RGVAR=IVAR(1)
      IF(MBPV(N).EQ.4)RGVAR=RVAR
      RETURN
      END
C
C
      FUNCTION RPVAR(N,ISUB,R)
      COMMON L,LF,LR,MBPV(11),MBASE(11)
      RPVAR=0
      LOC=IGLOC(N,ISUB)
      IF(MBPV(N).NE.2)GOTO 10
      I=R
D     TYPE 100,N,ISUB,R,LOC,L
D100  FORMAT(' N,ISUB,R,LOC,L',2I6,F15.7,2O7)
      CALL IPOKE(LOC,I)
      RETURN
10    IF(MBPV(N).NE.4)GOTO 20
      LOC1=IADDR(R)
D     TYPE 100,N,ISUB,R,LOC,L
      CALL IPOKE(LOC,IPEEK(LOC1))
      CALL IPOKE(LOC+2,IPEEK(LOC1+2))
      RETURN
20    RPVAR=-1.
      RETURN
C
C
      END
C
C
      FUNCTION ICUR(J)
C     THIS FUNCTION POSTIONS THE CURSOR TO BOTTOM OF PAGE
C     CLEARS THE LINE
C     FOR INTERACTIVE USE
      CALL ITTOUT("17,J,"0)
      DO 10 I=1,30
      CALL ITTOUT("40)
10    CONTINUE
      CALL ITTOUT("17,J,0)
      RETURN
      END

      SUBROUTINE IYX(IX,IY)
      CALL ITTOUT(15,IX,0)
      DO 20 I=1,IY
      CALL ITTOUT(9)
20    CONTINUE
      RETURN
      END
C
C
      FUNCTION UPDATE(IY,IX,VNEW,VOLD)
```

```
        UPDATE=0.
        IF(VNEW.EQ.VOLD)RETURN
        UPDATE=1.
        VOLD=VNEW
        CALL IYX(IY,IX-1)
        TYPE 100,INT(VNEW)
100     FORMAT('$',I5)
        RETURN
        END
*
```

```
          BLOCKDATA
C         COMMON DEFINITIONS
          COMMON IN(2),IOUT(2)
          COMMON IR(12),IC(12),IO(12)
          COMMON RE(12),ALPHA(12),BETA(12),TEMPX(12)
          COMMON ISTAT1(12),ISTAT2(12),JREF(12)
          COMMON MIN(12),MAX(12),SLOPE(12),BINTER(12)
          COMMON ICYCLE(281),NPTS(281),IREF(11,281)
C
C         DATA DECLARATIONS
          DATA IN/2*0/,IOUT/2*0/
          DATA IR/12*0/,IC/12*0/,IO/12*0/
          DATA RE/12*0./,ALPHA/12*0./,BETA/12*0./,TEMPX/12*0./
          DATA ISTAT1/12*0/,ISTAT2/12*0/,JREF/12*0/
          DATA MIN/12*0/
C         LEVEL 15 MAXIMUMS FOR POT CONTROL
          DATA MAX/
          1 1187,-2709,514,-449,1373,2,-2767,-1503,601,-2767,-2734,1/
          DATA SLOPE/12*0./,BINTER/12*0./
*
```

```
      BLOCKDATA
C     COMMON DEFINITIONS
      COMMON IN(2),IOUT(2)
      COMMON IR(12),IC(12),IO(12)
      COMMON RE(12),ALPHA(12),BETA(12),TEMPX(12)
      COMMON ISTAT1(12),ISTAT2(12),JREF(12)
      COMMON MIN(12),MAX(12),SLOPE(12),BINTER(12)
      COMMON ICYCLE(281),NPTS(281),IREF(11,281)
C
C     DATA DECLARATIONS
      DATA IN/2*0/,IOUT/2*0/
      DATA IR/12*0/,IC/12*0/,IO/12*0/
      DATA RE/12*0./,ALPHA/12*0./,BETA/12*0./,TEMPX/12*0./
      DATA ISTAT1/12*0/,ISTAT2/12*0/,JREF/12*0/
      DATA MIN/12*0/
C     LEVEL 15 MAXIMUMS FOR POT CONTROL
      DATA MAX/
     1 1187,-2709,514,-449,1373,2,-2767,-1503,601,-2767,-2734,1/
      DATA SLOPE/12*0./,BINTER/12*0./
      DATA ICYCLE/
     5 -2,-2,-1,-1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,1,
     5 1,-2,-2,-1,-1,1,1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,-2,-2,-1,-1,1,1,
     5 1,1,1,1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,-2,-2,-1,-1,1,1,-2,-2,-1,
     1 -1,1,1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,1,
     6 1,-2,-2,-1,-1,1,1,1,1,1,1,1,1,-2,-2,-1,-1,1,1,-2,-2,-1,-1,1,1,
     1 -2,-2,-1,-1,1,1,1,1,1,1,1,1,1,21,1,57,1,1,-2,-2,-1,-1,1,1,-2,
     5 -2,-1,-1,1,1,1,1,1,1,-2,-2,-1,-1,1,1,-2,-2,-1,-1,1,1,-2,-2,-1,
     5 -1,1,1,1,1,1,11,1,15,1,1,1,1,-2,-2,-1,-1,1,1,-2,-2,-1,-1,1,1,
     5 -2,-2,-1,-1,1,1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,-2,-2,-1,-1,1,1,1,
     4 1,1,1,-2,-2,-1,-1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
     5 1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,
     1 -1,-1,-1,-1,-1,1,/
      DATA NPTS/
     3 4184,2830,3959,2379,3754,1969,1230,1230,2256,2707,
     7 2461,2195,2010,1805,1394,1394,6310,5251,5825,3610,
     1 5518,2297,1764,205,1128,205,1764,3630,1292,2646,
     5 2215,1866,1497,492,205,697,2810,4492,3754,3015,
     3 2400,1784,1477,1148,2379,3466,3077,2728,2338,2030,
     7 1579,615,205,676,676,205,2112,205,2441,4718,
     1 4000,3302,2666,2153,1600,512,5046,4513,4307,3159,
     5 3610,1969,3425,4656,3938,3261,2625,2112,2030,205,
     9 2728,205,1497,3405,3138,2728,2318,1989,1723,574,
     8 205,820,2912,3323,2933,2564,2215,1887,1435,615,
     1 205,533,2379,3630,3138,2646,2215,1866,1497,492,
     5 307,205,2195,205,1374,3302,2954,2584,2318,2051,
     8 3425,3610,3220,2851,2543,2256,3610,3323,3282,2666,
     2 2995,2112,2584,3302,2461,2461,1948,3036,2871,2789,
     5 2482,2441,2112,2010,2995,5641,5087,4492,4000,3507,
     9 2523,1538,2461,1394,2359,1210,533,205,2297,205,
     2 1107,2461,2195,1928,1682,1435,2789,2707,2502,2153,
     7 2236,1600,2912,2523,2625,1928,2379,1435,1230,2830,
     2 2605,2502,2133,1928,861,205,1682,205,1723,
     8 3323,2851,2543,2153,2051,3713,3179,3302,2502,2912,
     1 1805,3630,3323,3692,2543,3774,2051,3836,205,1661,
     1 205,2769,5538,4779,4020,3425,2871,2297,1661,2830,
```

```
3 3795,3220,2954,2461,2174,1661,553,2520,205,4533,
7 4287,4369,1907,4205,1559,943,943,4574,2174,779,
1 205,1559,205,861,1866,1005,205,1661,205,1169,
5 2174,1661,205,4205,1559,943,943,2071,1559,1210,
9 943,4574,2174,1189,205,1559,205,861,1866,1005,
3 205,1661,205,1169,2318,1661,205,4205,1559,943,943,2023,/
DATA IREF/
1 2218,-4095,985,-753,-8,0,-2860,-2211,1014,-3240,-2981,
1 719,-1329,319,-245,-3,0,-929,-718,328,-1052,-968,
1 2100,-3879,933,-713,-8,0,-2709,-2094,960,-3068,-2823,
1 837,-1546,371,-284,-3,0,-1080,-835,382,-1223,-1126,
1 1983,-3662,880,-673,-7,0,-2557,-1977,907,-2897,-2666,
1 940,-1736,417,-319,-4,0,-1213,-937,429,-1373,-1264,
1 1455,-2715,644,-477,1200,0,-1894,-1465,671,-2146,-1975,
1 1469,-2712,652,-499,-6,0,-1894,-1465,671,-2146,-1975,
1 1771,-4001,786,-708,-7,0,-4095,-2224,890,-4095,-4046,
1 622,-1406,276,-249,-3,0,-1439,-782,312,-1439,-1422,
1 1664,-3758,738,-665,-7,0,-3846,-2088,836,-3846,-3800,
1 730,-1649,323,-292,-3,0,-1688,-917,366,-1688,-1668,
1 1580,-3569,701,-632,-7,0,-3653,-1983,794,-3653,-3609,
1 814,-1839,361,-326,-4,0,-1882,-1022,408,-1882,-1859,
1 1187,-2709,514,-449,1373,0,-2767,-1503,601,-2767,-2734,
1 1197,-2704,531,-479,-5,0,-2767,-1503,601,-2767,-2734,
1 3945,3476,2688,169,3041,0,-1339,-2597,1009,0,-1384,
1 215,809,515,795,-2081,0,-328,-254,342,0,802,
1 3340,3044,2336,271,2210,0,-1175,-2217,901,0,-1030,
1 769,1206,838,702,-1320,0,-479,-602,441,0,477,
1 2836,2683,2042,355,1518,0,-1038,-1901,811,0,-734,
1 1198,1512,1088,630,-732,0,-595,-871,518,0,226,
1 2125,2182,1631,477,997,0,-847,-1457,685,0,-321,
1 2125,2182,1631,477,997,0,-847,-1457,685,0,-321,
1 2174,1062,1794,354,706,163,-229,-1443,713,0,-1086,
1 2174,1062,1794,354,706,163,-229,-1443,713,0,-1086,
1 3912,1904,2904,490,855,195,-437,-2854,974,0,-2279,
1 366,174,633,205,-253,-58,-12,35,439,0,163,
1 3443,1675,2603,452,708,162,-381,-2472,903,0,-1957,
1 855,413,946,244,-100,-23,-71,-363,512,0,-173,
1 3035,1476,2342,420,581,133,-331,-2140,842,0,-1676,
1 1201,582,1168,272,8,1,-112,-645,565,0,-412,
1 2668,1297,2107,390,466,106,-288,-1841,786,0,-1423,
1 2179,1059,1794,351,314,71,-229,-1443,713,0,-1086,
1 2179,1059,1794,351,314,71,-229,-1443,713,0,-1086,
1 1487,721,1350,295,97,22,-146,-878,608,0,-609,
1 3820,1496,3564,187,1486,340,709,-1525,128,0,-3367,
1 231,1071,513,70,-1254,-287,-1171,-1530,980,0,1020,
1 3229,1426,3062,168,1035,236,399,-1526,269,0,-2645,
1 822,1141,1016,89,-803,-184,-861,-1529,840,0,297,
1 2743,1369,2649,152,664,152,145,-1527,384,0,-2051,
1 1308,1199,1429,105,-432,-99,-607,-1528,724,0,-296,
1 2490,1339,2434,144,471,107,12,-1527,444,0,-1742,
1 1561,1229,1644,113,-239,-55,-474,-1528,664,0,-606,
1 3866,2417,3589,118,445,101,-443,-3138,786,0,-2519,
1 476,329,734,137,-161,-37,-52,-172,359,0,-41,
1 3478,2179,3262,120,376,86,-398,-2799,737,0,-2235,
1 805,532,1011,135,-102,-24,-90,-460,400,0,-282,
1 3091,1940,2936,122,306,70,-354,-2460,688,0,-1952,
```

```
1 1096,711,1256,134,-50,-12,-124,-714,437,0,-494,
1 2645,1666,2561,125,227,51,-302,-2070,632,0,-1627,
1 2026,1284,2039,128,116,26,-231,-1528,554,0,-1174,
1 2026,1284,2039,128,116,26,-231,-1528,554,0,-1174,
1 1348,866,1468,132,-5,-2,-153,-934,469,0,-678,
1 2021,1287,2039,131,429,99,-231,-1528,554,0,-1174,
1 2021,1287,2039,131,429,99,-231,-1528,554,0,-1174,
1 -50,-63,709,1749,0,-191,570,-1372,843,0,-806,
1 -50,-63,709,1749,0,-191,570,-1372,843,0,-806,
1 14,-66,2238,4095,0,-969,1194,-3755,1946,0,-2209,
1 -102,-65,-726,-458,0,-540,-16,864,-191,0,510,
1 -3,-65,1779,3389,0,-903,1007,-3038,1614,0,-1787,
1 -85,-65,-295,205,0,-602,160,192,119,0,114,
1 -19,-65,1375,2769,0,-844,842,-2410,1323,0,-1417,
1 -72,-65,24,695,0,-648,290,-305,349,0,-178,
1 -33,-65,1038,2250,0,-795,704,-1884,1080,0,-1107,
1 -45,-65,709,1746,0,-748,570,-1372,843,0,-806,
1 -120,-139,867,3326,0,-1194,1117,-2205,-4095,0,129,
1 -85,-78,-367,309,0,-840,115,172,320,0,133,
1 -115,-129,665,2832,0,-1136,953,-1816,-3373,0,130,
1 -90,-86,-201,715,0,-887,250,-148,-274,0,132,
1 -109,-120,472,2360,0,-1081,796,-1445,-2682,0,130,
1 -94,-93,-66,1044,0,-926,359,-407,-756,0,132,
1 -11,-34,2248,4088,0,-1327,1189,-3763,1931,0,-2193,
1 -138,-38,-846,-390,0,-3949,-68,792,-325,0,650,
1 -31,-34,1768,3394,0,-1733,994,-3056,1581,0,-1753,
1 -120,-37,-395,262,0,-3567,115,129,3,0,236,
1 -48,-35,1347,2784,0,-2090,823,-2437,1274,0,-1366,
1 -106,-37,-62,744,0,-3285,250,-361,245,0,-70,
1 -41,-68,709,1743,0,-1304,570,-1372,843,0,-806,
1 -41,-68,709,1743,0,-1304,570,-1372,843,0,-806,
1 2623,616,1427,554,-256,-60,-191,-507,-940,0,-1129,
1 2623,616,1427,554,-256,-60,-191,-507,-940,0,-1129,
1 4095,1311,2370,682,251,57,-317,-958,-1778,0,-2062,
1 773,-245,248,402,-187,-43,-34,57,107,0,38,
1 3834,1189,2204,660,217,49,-294,-878,-1630,0,-1897,
1 1163,-62,498,434,-135,-31,-67,-62,-114,0,-209,
1 3443,1006,1954,627,165,37,-261,-759,-1408,0,-1650,
1 1489,91,706,462,-92,-22,-95,-162,-299,0,-415,
1 3183,884,1788,605,131,30,-239,-679,-1260,0,-1485,
1 2618,619,1427,557,56,13,-191,-507,-940,0,-1129,
1 2618,619,1427,557,56,13,-191,-507,-940,0,-1129,
1 1815,243,914,489,-50,-12,-123,-261,-484,0,-621,
1 3916,2989,3099,574,141,32,-911,-3113,1001,0,-3004,
1 660,44,427,322,-197,-45,-86,-121,386,0,-51,
1 3523,2633,2776,544,100,23,-812,-2751,926,0,-2647,
1 1018,368,721,350,-160,-37,-177,-450,453,0,-376,
1 3165,2309,2483,516,63,14,-721,-2423,859,0,-2323,
1 1322,643,970,373,-128,-30,-254,-729,511,0,-652,
1 2735,1921,2130,483,19,4,-612,-2028,778,0,-1933,
1 2127,1371,1631,436,-45,-11,-458,-1469,663,0,-1382,
1 2127,1371,1631,436,-45,-11,-458,-1469,663,0,-1382,
1 1590,886,1190,394,-100,-23,-322,-976,561,0,-895,
1 3912,1904,2904,490,855,195,-437,-2854,974,0,-2279,
1 366,174,633,205,-253,-58,-12,35,439,0,163,
```

```
1 3443,1675,2603,452,708,162,-381,-2472,903,0,-1957,
1 855,413,946,244,-100,-23,-71,-363,512,0,-173,
1 3035,1476,2342,420,581,133,-332,-2140,842,0,-1676,
1 1201,582,1168,272,8,1,-112,-645,565,0,-412,
1 2668,1297,2107,390,466,106,-288,-1841,786,0,-1423,
1 2179,1059,1794,351,314,71,-229,-1443,713,0,-1086,
1 2175,1062,1794,353,622,143,-229,-1443,713,0,-1086,
1 2175,1062,1794,353,622,143,-229,-1443,713,0,-1086,
1 23,103,1549,936,0,2202,1024,-872,1773,0,-1682,
1 23,103,1549,936,0,2202,1024,-872,1773,0,-1682,
1 -55,-1,2406,739,0,2497,2240,471,1115,0,-2728,
1 2,34,285,-69,0,-139,-633,-603,2031,0,504,
1 -49,2,2178,652,0,2213,1931,355,1214,0,-2380,
1 -4,30,513,18,0,144,-324,-487,1933,0,156,
1 -44,5,1998,583,0,1990,1687,264,1291,0,-2107,
1 -8,27,677,80,0,347,-103,-405,1862,0,-93,
1 -4,81,2708,1374,0,3093,2593,-285,1272,0,-3447,
1 58,119,391,492,0,214,-545,-1458,2273,0,83,
1 2,85,2463,1281,0,2789,2262,-409,1378,0,-3074,
1 52,115,636,585,0,518,-214,-1334,2167,0,-290,
1 8,88,2267,1206,0,2546,1996,-509,1462,0,-2776,
1 47,112,815,653,0,741,29,-1243,2090,0,-563,
1 73,200,2895,2174,0,3637,2770,-1463,1593,0,-4095,
1 131,235,758,1360,0,980,-126,-2545,2517,0,-839,
1 79,203,2683,2093,0,3373,2482,-1571,1685,0,-3772,
1 125,232,970,1441,0,1244,161,-2438,2425,0,-1162,
1 84,206,2504,2025,0,3150,2239,-1661,1762,0,-3499,
1 120,229,1150,1509,0,1467,404,-2347,2347,0,-1435,
1 240,435,2365,3325,0,3554,1907,-4095,2569,0,-3961,
1 27,100,1549,933,0,1654,1024,-872,1773,0,-1682,
1 187,351,2161,2727,0,3079,1686,-3289,2370,0,-3391,
1 27,100,1549,933,0,1654,1024,-872,1773,0,-1682,
1 153,298,2031,2344,0,2775,1545,-2774,2242,0,-3026,
1 -43,-10,1280,144,0,1027,733,192,1510,0,-930,
1 142,281,1990,2224,0,2680,1501,-2613,2202,0,-2912,
1 -38,-2,1301,203,0,1074,755,111,1530,0,-987,
1 123,251,1916,2009,0,2509,1421,-2322,2131,0,-2707,
1 -35,3,1313,239,0,1103,768,63,1541,0,-1021,
1 102,217,1835,1770,0,2319,1333,-2000,2051,0,-2479,
1 -28,13,1337,311,0,1160,794,-34,1565,0,-1089,
1 -169,-173,1101,2406,0,4095,2327,-964,922,0,-3643,
1 -103,-126,-1526,1900,0,-1418,-659,1097,-709,0,-814,
1 -162,-169,840,2356,0,3547,2030,-759,760,0,-3362,
1 -110,-131,-1249,1953,0,-836,-343,879,-536,0,-1113,
1 -157,-164,611,2312,0,3067,1771,-580,618,0,-3116,
1 -116,-135,-1020,1997,0,-356,-84,700,-395,0,-1359,
1 -83,-122,-14,2634,0,2112,944,-316,-586,0,-2787,
1 -190,-178,-395,1675,0,598,743,436,810,0,-1688,
1 -87,-124,-28,2597,0,2054,936,-287,-533,0,-2745,
1 -184,-175,-376,1724,0,676,754,398,739,0,-1744,
1 -94,-128,-53,2535,0,1957,923,-239,-443,0,-2674,
1 -177,-171,-351,1786,0,773,767,349,649,0,-1815,
1 -132,-153,-204,2151,0,814,844,60,112,0,-2238,
```

```
1 -132,-153,-204,2151,0,814,844,60,112,0,-2238,
1 155,519,2039,525,-4,-10,808,-396,-735,0,-777,
1 155,519,2039,525,-4,-10,808,-396,-735,0,-777,
1 90,443,2549,313,367,1085,1237,193,358,0,-927,
1 -160,245,142,168,-319,-944,-309,-65,-121,0,407,
1 62,421,2284,297,291,862,1067,165,306,0,-780,
1 -135,265,387,183,-250,-738,-152,-39,-72,0,271,
1 37,401,2039,282,221,655,910,138,257,0,-645,
1 -109,286,632,197,-180,-531,5,-13,-23,0,136,
1 288,631,3364,608,494,1461,1660,-254,-471,0,-1512,
1 13,413,713,448,-262,-775,-44,-538,-999,0,-42,
1 259,607,3079,590,412,1220,1476,-284,-527,0,-1354,
1 41,435,978,464,-187,-551,126,-510,-947,0,-189,
1 231,586,2814,574,337,996,1306,-313,-580,0,-1207,
1 68,456,1243,480,-111,-328,297,-481,-894,0,-336,
1 501,836,4095,946,575,1701,2007,-816,-1516,0,-2075,
1 245,633,1627,797,-129,-380,421,-1081,-2007,0,-707,
1 472,812,3809,929,494,1461,1824,-847,-1572,0,-1917,
1 275,656,1912,814,-47,-139,605,-1051,-1951,0,-865,
1 446,792,3564,914,424,1254,1666,-873,-1621,0,-1781,
1 300,676,2157,829,23,68,762,-1024,-1902,0,-1001,
1 508,863,3370,1079,292,865,1468,-1279,-2374,0,-1769,
1 -95,288,1124,149,-6,-16,354,211,392,0,-95,
1 460,817,3190,1004,268,795,1379,-1159,-2152,0,-1635,
1 -76,305,1193,177,3,11,388,165,306,0,-147,
1 378,738,2885,878,228,675,1228,-957,-1776,0,-1407,
1 -35,344,1345,241,24,71,464,64,118,0,-260,
1 155,519,2039,525,-5,-14,808,-396,-735,0,-777,
1 155,519,2039,525,-5,-14,808,-396,-735,0,-777,
1 1813,925,1305,514,-217,-51,-50,-1556,503,0,-1164,
1 1813,925,1305,514,-217,-51,-50,-1556,503,0,-1164,
1 3387,1757,2076,831,485,111,-117,-3249,581,0,-2560,
1 363,169,598,228,-270,-62,12,-5,431,0,114,
1 2972,1539,1873,748,382,87,-99,-2803,560,0,-2192,
1 645,317,736,284,-199,-46,0,-308,445,0,-136,
1 2606,1347,1694,675,290,66,-83,-2411,542,0,-1869,
1 728,361,777,301,-178,-41,-4,-397,449,0,-209,
1 3369,1159,1839,725,587,134,417,-1716,-3187,0,-2011,
1 450,221,569,207,-138,-32,42,-40,-75,0,-26,
1 2998,1040,1677,659,495,113,369,-1503,-2791,0,-1759,
1 703,303,679,252,-75,-18,75,-186,-345,0,-198,
1 2627,921,1516,593,403,92,322,-1290,-2395,0,-1507,
1 973,389,796,300,-8,-2,109,-341,-632,0,-382,
1 3298,1834,2107,832,-3551,-813,-212,-3297,490,0,-2028,
1 277,242,628,229,-4095,-938,-79,-47,354,0,617,
1 2983,1616,1904,750,-3626,-830,-194,-2850,472,0,-1665,
1 559,391,766,286,-4045,-926,-91,-350,367,0,370,
1 2518,1423,1725,677,-3691,-845,-177,-2457,455,0,-1345,
1 642,435,807,302,-4030,-923,-95,-440,371,0,297,
1 1814,925,1305,513,-273,-64,-50,-1556,503,0,-1164,
1 1814,925,1305,513,-273,-64,-50,-1556,503,0,-1164,
1 1390,1897,2080,26,-1908,0,-560,-780,1944,0,-504,
1 1390,1897,2080,26,-1908,0,-560,-780,1944,0,-504,
1 3638,1789,3117,-75,532,0,9,-721,1613,0,-3210,
```

```
1 -869,2012,1042,132,-3516,0,-1129,-838,2275,0,2202,
1 3022,1819,2633,-46,-21,0,-147,-727,1703,0,-2478,
1 -252,1981,1326,104,-2962,0,-974,-822,2184,0,1462,
1 2533,1843,2608,-24,-460,0,-270,-750,1775,0,-1833,
1 194,1959,1531,84,-2561,0,-861,-810,2119,0,925,
1 2065,1867,2393,-2,-881,0,-388,-762,1844,0,-1321,
1 704,1934,1766,60,-2103,0,-732,-797,2044,0,313,
1 3477,2999,2752,98,-36,0,-1045,-2540,2169,0,-1759,
1 -241,1054,1558,-17,-2622,0,-183,587,1769,0,470,
1 2908,2693,2569,72,-431,0,-913,-2061,2107,0,-1418,
1 23,1192,1642,-10,-2439,0,-245,365,1798,0,312,
1 2441,2450,2419,59,-757,0,-805,-1668,2057,0,-1138,
1 309,1340,1734,-1,-2241,0,-311,126,1828,0,141,
1 1933,2186,2256,44,-1110,0,-687,-1241,2003,0,-833,
1 1385,1900,2080,29,-1492,0,-560,-780,1944,0,-504,
1 1805,1316,1631,332,-3861,0,-605,-822,1866,0,-227,
1 1805,1316,1631,332,-3861,0,-605,-822,1866,0,-227,
1 1736,-3116,351,-310,-5,0,-1559,-837,-1554,-2108,-2520,
1 518,-931,105,-93,-2,0,-466,-250,-464,-630,-753,
1 1646,-2954,333,-294,-5,0,-1478,-793,-1473,-1999,-2389,
1 608,-1093,123,-109,-2,0,-547,-294,-545,-740,-884,
1 1555,-2792,315,-278,-5,0,-1397,-750,-1392,-1889,-2259,
1 699,-1255,141,-125,-2,0,-628,-337,-626,-849,-1015,
1 1110,-2013,228,-189,921,0,-1013,-544,-1009,-1369,-1637,
1 1127,-2023,228,-201,-4,0,-1013,-544,-1009,-1369,-1637,
1 2441,2450,2419,59,-757,0,-805,-1668,2057,0,-1138,
1 308,1340,1734,-1,-2241,0,-311,126,1828,0,141,
1 1379,1904,2080,32,-1076,0,-560,-780,1944,0,-504,
1 1379,1904,2080,32,-1076,0,-560,-780,1944,0,-504,
1 1803,932,1305,520,455,105,-50,-1556,503,0,-1164,
1 1803,932,1305,520,455,105,-50,-1556,503,0,-1164,
1 2606,1347,1694,675,290,66,-83,-2411,542,0,-1869,
1 894,448,858,334,-137,-32,-11,-575,457,0,-356,
1 1814,925,1305,513,-273,-64,-50,-1556,503,0,-1164,
1 1814,925,1305,513,-273,-64,-50,-1556,503,0,-1164,
1 1390,1897,2080,26,-1908,0,-560,-780,1944,0,-504,
1 1390,1897,2080,26,-1908,0,-560,-780,1944,0,-504,
1 2441,2450,2419,59,-757,0,-805,-1668,2057,0,-1138,
1 308,1340,1734,-1,-2241,0,-311,126,1828,0,141,
1 1805,1316,1631,332,-3861,0,-605,-922,1856,0,-227,
1 1805,1316,1631,332,-3861,0,-605,-922,1866,0,-227,
1 1555,-2792,315,-278,-5,0,-1397,-750,-1392,-1999,-2259
1 699,-1255,141,-125,-2,0,-628,-337,-626,-849,-1015,
1 1110,-2013,228,-189,921,0,-1013,-544,-1009,-1369,-1637,
1 1127,-2023,228,-201,-4,0,-1013,-544,-1009,-1369,-1637,
1 1555,-2792,315,-278,-5,0,-1397,-750,-1392,-1889,-2259,
1 699,-1255,141,-125,-2,0,-628,-337,-626,-849,-1015,
1 1110,-2013,228,-189,921,0,-1013,-544,-1009,-1369,-1637,
1 1127,-2023,228,-201,-4,0,-1013,-544,-1009,-1369,-1637,
1 2441,2450,2419,59,-757,0,-805,-1668,2057,0,-1138,
1 308,1340,1734,-1,-2241,0,-311,126,1828,0,141,
1 1379,1904,2080,32,-1076,0,-560,-780,1944,0,-504,
1 1379,1904,2080,32,-1076,0,-560,-780,1944,0,-504,
1 1803,932,1305,520,455,105,-50,-1556,503,0,-1164,
```

```
1 1803,932,1305,520,455,105,-50,-1556,503,0,-1164,
1 2606,1247,1694,675,290,66,-83,-2411,542,0,-1869,
1 894,448,858,334,-137,-32,-11,-575,457,0,-356,
1 1814,925,1305,513,-273,-64,-50,-1556,503,0,-1164,
1 1814,925,1305,513,-273,-64,-50,-1556,503,0,-1164,
1 1390,1897,2080,26,-1908,0,-560,-780,1944,0,-504,
1 1390,1897,2080,26,-1908,0,-560,-780,1944,0,-504,
1 2441,2450,2419,59,-757,0,-805,-1668,2057,0,-1138,
1 166,1266,1688,-6,-2340,0,-278,246,1813,0,227,
1 1805,1316,1631,332,-3861,0,-605,-822,1866,0,-227,
1 1805,1316,1631,332,-3861,0,-605,-822,1866,0,-227,
1 1555,-2792,315,-278,-5,0,-1397,-750,-1392,-1889,-2259,
1 699,-1255,141,-125,-2,0,-628,-337,-626,-849,-1015,
1 1110,-2013,228,-189,921,0,-1013,-544,-1009,-1369,-1637,
1 1127,-2023,228,-201,-4,0,-1013,-544,-1009,-1369,-1637,
1 0,0,0,0,0,0,0,0,0,0,0,
1 /
END
```
*

```
 TT:CAMSGEN.BAS
10 REM MOD 30-NOV-76 INVERT ICYCLE ARRAY (11,281)
50 DIM L(281,11)
51 DIM C(281)
52 DIM P(11)
53 DIM D(281)
60 PRINT 'NUMBER OF DIFERENCE UNITS PER OUTPUT STEP (M)',
65 INPUT M
70 PRINT 'SCALE FACTOR',
75 INPUT S
100 OPEN 'CYCLE.DAT' FOR INPUT AS FILE #1
110 OPEN 'LOADS.DAT' FOR INPUT AS FILE #2
200 FOR I=1 TO 11
210 FOR J=1 TO 281
215 IF END #2GO TO 900
220 INPUT #2:L(J,I)
230 NEXT J
240 NEXT I
300 FOR I=1 TO 281
305 IF END #1GO TO 900
310 INPUT #1:C(I)
320 NEXT I
400 FOR I=1 TO 11\READ P(I)\NEXT I
500 REM SCALE THE TABLE
510 FOR I=1 TO 11
515 FOR J=1 TO 281
520 L(J,I)=INT(L(J,I)/P(I)*S)
530 NEXT J
540 NEXT I
600 REM DETERMINE NUMBER OF POINTS PER RAMP
610 FOR J=2 TO 281
612 T=M
614 FOR I=1 TO 11
620 IF T>ABS(L(J,I)-L(J-1,I)) THEN 630 \T=ABS(L(J,I)-L(J-1,I))
630 NEXT I
635 D(J)=INT(T/M)
640 NEXT J
650 D(1)=6200
660 CLOSE
800 REM OUTPUT THE DATA
805 OPEN 'TABLE.FOR' FOR OUTPUT AS FILE #3
810 PRINT #3:CHR$(9);'DATA ICYCLE/'
820 FOR I=1 TO 281
830 A=C(I)
840 GOSUB 2000
850 NEXT I
860 GOSUB 3000
900 PRINT #3:CHR$(9);'DATA NPTS/'
910 FOR I=1 TO 281\A=D(I)\GOSUB 2000 \NEXT I
920 GOSUB 3000
950 PRINT #3:CHR$(9);'DATA IREF/'
960 FOR J=1 TO 281\FOR I=1 TO 11
970 A=L(J,I)
980 GOSUB 2000
990 NEXT I
992 GOSUB 2060
994 NEXT J
999 GOSUB 3000
1100 PRINT #3:CHR$(9);'END'
```

```
1900 CLOSE
1910 OPEN 'BLOCKD.PRO' FOR INPUT AS FILE #1
1920 OPEN 'TABLE.FOR' FOR INPUT AS FILE #2
1930 OPEN 'BLOCKD.FOR' FOR OUTPUT AS FILE #3
1940 IF END #1 THEN 1960
1950 INPUT #1:A$\PRINT #3:A$\GO TO 1940
1960 IF END #2 THEN 1980
1970 INPUT #2:A$\PRINT #3:A$\GO TO 1960
1980 CLOSE
1999 GO TO 32000
2000 REM TAKE ANOTHER DATA POINT & BUILD STRING
2010 GOSUB 4000
2015 Z$=STR$(A)
2020 A$=A$&Z$&','
2025 IF LEN(A$)>60GO TO 2060
2030 RETURN
2060 PRINT #3:CHR$(9)&I$&' '&A$
2070 A$=''
2080 RETURN
3000 REM END OF SECTION
3010 PRINT #3:CHR$(9)&I$&' '&A$&'/'
3020 A$=''
3030 RETURN
4000 REM CONVERT VAR I TO SINGLE DIGIT STRING
4010 I$=STR$(I)
4020 I$=SEG$(I$,LEN(I$),LEN(I$))
4025 IF I$<>'0' THEN 4030 \I$='5'
4030 RETURN
10000 DATA 78304,77262,50200,116694,247430
10100 DATA 135957,129699,18234,8813
10200 DATA 89063,59929
32000 END
*
```

```
DATA ICYCLE/
5 -2,-2,-1,-1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,1,
5 1,-2,-2,-1,-1,1,1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,-2,-2,-1,-1,1,1,
5 1,1,1,1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,-2,-2,-1,-1,1,1,-2,-2,-1,
1 -1,1,1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,1,
6 1,-2,-2,-1,-1,1,1,1,1,1,1,1,1,-2,-2,-1,-1,1,1,-2,-2,-1,-1,1,1,
1 -2,-2,-1,-1,1,1,1,1,1,1,1,1,1,21,1,57,1,1,-2,-2,-1,-1,1,1,-2,
5 -2,-1,-1,1,1,1,1,1,1,-2,-2,-1,-1,1,1,-2,-2,-1,-1,1,1,-2,-2,-1,
5 -1,1,1,1,1,1,11,1,15,1,1,1,1,-2,-2,-1,-1,1,1,-2,-2,-1,-1,1,1
7 -2,-2,-1,-1,1,1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,-2,-2,-1,-1,1,1,1,
4 1,1,1,-2,-2,-1,-1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
5 1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,
1 -1,-1,-1,-1,-1,1,/
DATA NPTS/
3 6200,2766,2558,[illegible],2116,1926,1204,1206,2201,2656,2407,2158,19

7 1771,1377,1378,6180,5122,4291,3530,2838,2250,1729,1,1122,1,173

1 3546,3077,2588,2180,1834,1467,489,1,692,2758,4237,3665,2942,23

5 1755,1446,1136,2305,3390,3002,2673,2236,1995,1549,619,1,678,67

9 1,2071,1,2393,4619,3902,3220,2602,2105,1579,512,4938,4415,3693

2 3099,2408,1926,3355,4555,3848,3185,2566,2076,1981,1,2664,1,147

7 [illegible],[illegible],2671,2290,1954,1694,565,1,803,2852,3256,2863,2505,21

1 1843,1413,608,1,537,2322,3546,3077,2588,2180,1834,1467,489,308

5 1,2152,1,1343,3232,2884,2536,2263,2014,3354,3530,3157,2784,248

8 2213,3532,3256,2933,2610,2307,2064,2526,3223,2417,2417,1902,29

1 2805,2724,2433,2385,2063,1966,2935,5513,4965,4383,3903,3423,24

5 1514,1456,1378,1281,1184,537,1,2243,1,1095,2407,2142,1897,1652

9 1407,2732,2651,2366,2101,1836,1571,2852,2468,2182,1897,1652,14

2 1213,2766,2544,2458,2082,1894,853,1,1659,1,1693,3244,2798,2495

5 2103,2014,3626,3112,2716,2446,2050,1763,2543,3250,2803,2500,21

9 2017,3757,1,1635,1,2706,5412,4672,3932,3345,2808,2246,1634,277

2 3718,3149,2335,2418,2133,1625,548,2359,1,4432,2185,2023,1861,
7 1599,1537,923,925,4473,2133,1165,1,1531,1,855,1836,981,1,1635,
2 1,1151,2133,1620,1,4108,1537,923,925,769,1537,923,925,4473,213

9 1165,1,1531,1,855,1836,981,1,1635,1,1151,2275,1639,1,4108,1537

1 923,925,2023,/
DATA IREF/
1 2218,-4095,985,-753,-8,0,-2860,-2211,1014,-3240,-2981,
```

```
1 719,-1319,319,-245,-3,0,-929,-718,328,-1052,-968,
1 2100,-3879,933,-713,-8,0,-2709,-2094,960,-3068,-2823,
1 837,-1546,371,-284,-3,0,-1080,-835,382,-1223,-1126,
1 1983,-3662,880,-673,-7,0,-2557,-1977,927,-2897,-2666,
1 940,-1735,417,-319,-4,0,-1213,-937,429,-1373,-1264,
1 1455,-2715,644,-477,1239,0,-1894,-1465,671,-2146,-1975,
1 1459,-2712,651,-499,-6,0,-1894,-1465,671,-2146,-1975,
1 1771,-4001,786,-708,-7,0,-4095,-2224,890,-4095,-4046,
1 622,-1406,276,-249,-3,0,-1439,-782,312,-1439,-1422,
1 1664,-3758,738,-665,-7,0,-3846,-2088,836,-3846,-3800,
1 730,-1649,323,-292,-3,0,-1688,-917,366,-1688,-1668,
1 1580,-3569,701,-632,-7,0,-3653,-1983,794,-3653,-3609,
1 814,-1839,361,-326,-4,0,-1882,-1022,408,-1882,-1859,
1 1187,-2709,514,-449,1373,0,-2767,-1503,601,-2767,-2734,
1 1197,-2704,531,-479,-5,0,-2767,-1503,601,-2767,-2734,
1 3945,3476,2688,169,3041,0,-1339,-2597,1009,0,-1384,
1 215,809,515,795,-2081,0,-328,-254,342,0,802,
1 3340,3044,2336,271,2210,0,-1175,-2217,901,0,-1030,
1 769,1206,838,702,-1320,0,-479,-602,441,0,477,
1 2836,2683,2042,355,1518,0,-1038,-1901,811,0,-734,
1 1198,1512,1088,630,-732,0,-595,-871,518,0,226,
1 2125,2182,1631,477,997,0,-847,-1457,685,0,-321,
1 2125,2182,1631,477,997,0,-847,-1457,685,0,-321,
1 2174,1062,1794,354,706,163,-229,-1443,713,0,-1086,
1 2174,1062,1794,354,706,163,-229,-1443,713,0,-1086,
1 3912,1904,2904,490,855,195,-437,-2854,974,0,-2279,
1 366,174,633,205,-253,-58,-12,35,439,0,163,
1 3443,1675,2603,452,708,162,-381,-2472,903,0,-1957,
1 855,413,946,244,-100,-23,-71,-363,512,0,-173,
1 3035,1476,2342,420,581,133,-331,-2140,842,0,-1676,
1 1201,582,1168,272,8,1,-112,-645,565,0,-412,
1 2668,1297,2107,390,466,106,-288,-1841,786,0,-1423,
1 2179,1059,1794,351,314,71,-229,-1443,713,0,-1086,
1 2179,1059,1794,351,314,71,-229,-1443,713,0,-1086,
1 1487,721,1350,295,97,22,-146,-878,608,0,-609,
1 3820,1496,3564,187,1486,340,709,-1525,128,0,-3367,
1 231,1071,513,70,-1254,-287,-1171,-1530,980,0,1020,
1 3229,1426,3062,168,1035,236,399,-1526,269,0,-2645,
1 822,1141,1016,89,-803,-184,-861,-1529,840,0,297,
1 2743,1369,2649,152,664,152,145,-1527,384,0,-2051,
1 1308,1199,1429,105,-432,-99,-607,-1528,724,0,-296,
1 2490,1339,2434,144,471,107,12,-1527,444,0,-1742,
1 1561,1229,1644,113,-239,-55,-474,-1528,664,0,-606,
1 3866,2417,3589,118,445,101,-443,-3138,786,0,-2519,
1 476,329,734,137,-161,-37,-52,-172,359,0,-41,
1 3478,2179,3262,120,376,86,-398,-2799,737,0,-2235,
1 805,532,1011,135,-102,-24,-90,-460,400,0,-282,
1 3091,1940,2936,122,306,70,-354,-2460,688,0,-1952,
1 1096,711,1256,134,-50,-12,-124,-714,437,0,-494,
1 2645,1666,2561,125,227,51,-302,-2070,632,0,-1627,
1 2026,1284,2039,128,116,26,-231,-1528,554,0,-1174,
1 2026,1284,2039,128,116,26,-231,-1528,554,0,-1174,
1 1348,866,1468,132,-5,-2,-153,-934,469,0,-678,
1 2021,1297,2039,131,429,99,-231,-1528,554,0,-1174.
```

```
1 2021,1287,2039,131,429,99,-231,-1528,554,0,-1174,
1 -50,-63,709,1749,0,-191,570,-1372,843,0,-806,
1 -50,-63,709,1749,0,-191,570,-1372,843,0,-806,
1 14,-66,2238,4095,0,-969,1194,-3755,1946,0,-2209,
1 -102,-65,-726,-458,0,-540,-16,864,-191,0,510,
1 -3,-65,1779,3389,0,-903,1007,-3038,1614,0,-1787,
1 -85,-65,-295,205,0,-602,160,192,119,0,114,
1 -19,-65,1375,2769,0,-844,842,-2410,1323,0,-1417,
1 -72,-65,24,695,0,-648,290,-305,349,0,-178,
1 -33,-65,1038,2250,0,-795,704,-1884,1080,0,-1107,
1 -45,-65,709,1746,0,-748,570,-1372,843,0,-806,
1 -120,-139,867,3326,0,-1194,1117,-2205,-4095,0,129,
1 -85,-78,-367,309,0,-840,115,172,320,0,133,
1 -115,-129,665,2832,0,-1136,953,-1816,-3373,0,130,
1 -90,-86,-201,715,0,-887,250,-148,-274,0,132,
1 -109,-120,472,2360,0,-1081,796,-1445,-2682,0,130,
1 -94,-93,-66,1044,0,-926,359,-407,-756,0,132,
1 -11,-34,2248,4088,0,-1327,1189,-3763,1931,0,-2193,
1 -138,-38,-846,-390,0,-3949,-68,792,-325,0,650,
1 -31,-34,1768,3394,0,-1733,994,-3056,1581,0,-1753,
1 -120,-37,-395,262,0,-3567,115,129,3,0,236,
1 -48,-35,1347,2784,0,-2090,823,-2437,1274,0,-1366,
1 -106,-37,-62,744,0,-3285,250,-361,245,0,-70,
1 -41,-68,709,1743,0,-1304,570,-1372,843,0,-806,
1 -41,-68,709,1743,0,-1304,570,-1372,843,0,-806,
1 2623,616,1427,554,-256,-60,-191,-507,-940,0,-1129,
1 2623,616,1427,554,-256,-60,-191,-507,-940,0,-1129,
1 4095,1311,2370,682,251,57,-317,-958,-1778,0,-2062,
1 773,-245,248,402,-187,-43,-34,57,107,0,38,
1 3834,1189,2204,660,217,49,-294,-878,-1630,0,-1897,
1 1163,-62,498,434,-135,-31,-67,-62,-114,0,-209,
1 3443,1006,1954,627,165,37,-261,-759,-1408,0,-1650,
1 1499,91,706,462,-92,-22,-95,-162,-299,0,-415,
1 3193,884,1788,605,131,30,-239,-679,-1260,0,-1485,
1 2618,619,1427,557,56,13,-191,-507,-940,0,-1129,
1 2618,619,1427,557,56,13,-191,-507,-940,0,-1129,
1 1815,243,914,489,-50,-12,-123,-261,-484,0,-621,
1 3916,2989,3099,574,141,32,-911,-3113,1001,0,-3004,
1 660,44,427,322,-197,-45,-86,-121,386,0,-51,
1 3523,2633,2776,544,100,23,-812,-2751,926,0,-2647,
1 1018,368,721,350,-160,-37,-177,-450,453,0,-376,
1 3165,2309,2483,516,63,14,-721,-2423,859,0,-2323,
1 1322,643,970,373,-128,-30,-254,-729,511,0,-652,
1 2735,1921,2130,483,19,4,-612,-2028,778,0,-1933,
1 2127,1371,1631,436,-45,-11,-458,-1469,663,0,-1382,
1 2127,1371,1631,436,-45,-11,-458,-1469,663,0,-1382,
1 1590,886,1190,394,-100,-23,-322,-976,561,0,-895,
1 3912,1904,2904,490,855,195,-437,-2854,974,0,-2279,
1 366,174,633,205,-253,-58,-12,35,439,0,163,
1 3443,1675,2603,452,708,162,-381,-2472,903,0,-1957,
1 855,413,946,244,-100,-23,-71,-363,512,0,-173,
1 3035,1476,2342,420,581,133,-332,-2140,842,0,-1676,
1 1201,582,1168,272,8,1,-112,-645,565,0,-412,
1 2668,1297,2107,390,466,106,-288,-1841,786,0,-1423,
1 2179,1059,1794,351,314,71,-229,-1443,713,0,-1086,
```

```
1 2175,1062,1794,353,622,143,-229,-1443,713,0,-1086,
1 2175,1062,1794,353,622,143,-229,-1443,713,0,-1086,
1 23,103,1549,936,0,2202,1024,-872,1773,0,-1682,
1 23,103,1549,936,0,2202,1024,-872,1773,0,-1682,
1 -55,-1,2406,739,0,2497,2240,471,1115,0,-2728,
1 2,34,285,-69,0,-139,-633,-603,2031,0,504,
1 -49,2,2178,652,0,2213,1931,355,1214,0,-2380,
1 -4,30,513,18,0,144,-324,-487,1933,0,156,
1 -44,5,1998,583,0,1990,1687,264,1291,0,-2107,
1 -8,27,677,80,0,347,-103,-405,1862,0,-93,
1 -4,81,2708,1374,0,3093,2593,-285,1272,0,-3447,
1 58,119,391,492,0,214,-545,-1458,2273,0,83,
1 2,85,2463,1281,0,2789,2262,-409,1378,0,-3074,
1 52,115,636,585,0,518,-214,-1334,2167,0,-290,
1 8,88,2267,1206,0,2546,1996,-509,1462,0,-2776,
1 47,112,815,653,0,741,29,-1243,2090,0,-563,
1 73,200,2895,2174,0,3637,2770,-1463,1593,0,-4095,
1 131,235,758,1360,0,980,-126,-2545,2517,0,-839,
1 79,203,2683,2093,0,3373,2482,-1571,1685,0,-3772,
1 125,232,970,1441,0,1244,161,-2438,2425,0,-1162,
1 84,206,2504,2025,0,3150,2239,-1661,1762,0,-3499,
1 120,229,1150,1509,0,1467,404,-2347,2347,0,-1435,
1 240,435,2365,3325,0,3554,1907,-4095,2569,0,-3961,
1 27,100,1549,933,0,1654,1024,-872,1773,0,-1682,
1 187,351,2161,2727,0,3079,1686,-3289,2370,0,-3391,
1 27,100,1549,933,0,1654,1024,-872,1773,0,-1682,
1 153,298,2031,2344,0,2775,1545,-2774,2242,0,-3026,
1 -43,-10,1280,144,0,1027,733,192,1510,0,-930,
1 142,281,1990,2224,0,2680,1501,-2613,2202,0,-2912,
1 -38,-2,1301,203,0,1074,755,111,1530,0,-987,
1 123,251,1916,2009,0,2509,1421,-2322,2131,0,-2707,
1 -35,3,1313,239,0,1103,768,63,1541,0,-1021,
1 102,217,1835,1770,0,2319,1333,-2000,2051,0,-2479,
1 -28,13,1337,311,0,1160,794,-34,1565,0,-1089,
1 -169,-173,1101,2406,0,4095,2327,-964,922,0,-3643,
1 -103,-126,-1526,1900,0,-1418,-659,1097,-709,0,-814,
1 -162,-169,840,2356,0,3547,2030,-759,760,0,-3362,
1 -110,-131,-1249,1953,0,-836,-343,879,-536,0,-1113,
1 -157,-164,611,2312,0,3067,1771,-580,618,0,-3116,
1 -116,-135,-1020,1997,0,-356,-84,700,-395,0,-1359,
1 -83,-122,-14,2634,0,2112,944,-316,-586,0,-2787,
1 -190,-178,-395,1675,0,598,743,436,810,0,-1688,
1 -87,-124,-28,2597,0,2054,936,-287,-533,0,-2745,
1 -184,-175,-376,1724,0,676,754,398,739,0,-1744,
1 -94,-128,-53,2535,0,1957,923,-239,-443,0,-2674,
1 -177,-171,-351,1786,0,773,767,349,649,0,-1815,
1 -132,-153,-204,2151,0,814,844,60,112,0,-2238,
1 -132,-153,-204,2151,0,814,844,60,112,0,-2238,
1 155,519,2039,525,-4,-10,808,-396,-735,0,-777,
1 155,519,2039,525,-4,-10,808,-396,-735,0,-777,
1 90,443,2549,313,367,1085,1237,193,358,0,-927,
1 -160,245,142,168,-319,-944,-309,-65,-121,0,407,
1 62,421,2284,297,291,862,1067,165,306,0,-780,
1 -135,265,387,183,-250,-738,-152,-39,-72,0,271,
1 37,401,2039,282,221,655,910,138,257,0,-645,
1 -109,286,632,197,-180,-531,5,-13,-23,0,136,
```

```
1 288,631,3364,608,494,1461,1660,-254,-471,0,-1512,
1 13,413,713,448,-262,-775,-44,-538,-999,0,-42,
1 259,607,3079,590,412,1220,1476,-284,-527,0,-1354,
1 41,435,978,464,-187,-551,126,-510,-947,0,-189,
1 231,586,2814,574,337,996,1306,-313,-580,0,-1207,
1 68,456,1243,480,-111,-328,297,-481,-894,0,-336,
1 501,836,4095,946,575,1701,2007,-816,-1516,0,-2075,
1 245,633,1627,797,-129,-380,421,-1081,-2007,0,-707,
1 472,812,3809,929,494,1461,1824,-847,-1572,0,-1917,
1 275,656,1912,814,-47,-139,605,-1051,-1951,0,-865,
1 446,792,3564,914,424,1254,1666,-873,-1621,0,-1781,
1 300,676,2157,829,23,68,762,-1024,-1902,0,-1001,
1 508,863,3370,1079,292,865,1468,-1279,-2374,0,-1769,
1 -95,288,1124,149,-6,-16,354,211,392,0,-95,
1 460,817,3190,1004,268,795,1379,-1159,-2152,0,-1635,
1 -76,305,1193,177,3,11,388,165,306,0,-147,
1 378,738,2885,878,228,675,1228,-957,-1776,0,-1407,
1 -35,344,1345,241,24,71,464,64,118,0,-260,
1 155,519,2039,525,-5,-14,808,-396,-735,0,-777,
1 155,519,2039,525,-5,-14,808,-396,-735,0,-777,
1 1813,925,1305,514,-217,-51,-50,-1556,503,0,-1164,
1 1813,925,1305,514,-217,-51,-50,-1556,503,0,-1164,
1 3387,1757,2076,831,485,111,-117,-3249,581,0,-2560,
1 363,169,598,228,-270,-62,12,-5,431,0,114,
1 2972,1539,1873,748,382,87,-99,-2803,560,0,-2192,
1 645,317,736,284,-199,-46,0,-308,445,0,-136,
1 2606,1347,1694,675,290,66,-83,-2411,542,0,-1869,
1 728,361,777,301,-178,-41,-4,-397,449,0,-209,
1 3369,1159,1839,725,587,134,417,-1716,-3187,0,-2011,
1 450,221,569,207,-138,-32,42,-40,-75,0,-26,
1 2998,1040,1677,659,495,113,369,-1503,-2791,0,-1759,
1 703,303,679,252,-75,-18,75,-186,-345,0,-198,
1 2627,921,1516,593,403,92,322,-1290,-2395,0,-1507,
1 973,389,796,300,-8,-2,109,-341,-632,0,-382,
1 3298,1834,2107,832,-3551,-813,-212,-3297,490,0,-2028,
1 277,242,628,229,-4095,-938,-79,-47,354,0,617,
1 2883,1616,1904,750,-3626,-830,-194,-2850,472,0,-1665,
1 559,391,766,286,-4045,-926,-91,-350,367,0,370,
1 2518,1423,1725,677,-3691,-845,-177,-2457,455,0,-1345,
1 642,435,807,302,-4030,-923,-95,-440,371,0,297,
1 1814,925,1305,513,-273,-64,-50,-1556,503,0,-1164,
1 1814,925,1305,513,-273,-64,-50,-1556,503,0,-1164,
1 1390,1897,2080,26,-1908,0,-560,-780,1944,0,-504,
1 1390,1897,2080,26,-1908,0,-560,-780,1944,0,-504,
1 3638,1789,3117,-75,532,0,9,-721,1613,0,-3210,
1 -869,2012,1042,132,-3516,0,-1129,-838,2275,0,2202,
1 3022,1819,2833,-46,-21,0,-147,-737,1703,0,-2470,
1 -252,1981,1326,104,-2962,0,-974,-822,2184,0,1462,
1 2533,1843,2608,-24,-460,0,-270,-750,1775,0,-1883,
1 194,1959,1531,84,-2561,0,-861,-810,2119,0,925,
1 2065,1867,2393,-2,-881,0,-388,-762,1844,0,-1321,
1 704,1934,1766,60,-2103,0,-732,-797,2044,0,313,
1 3477,2989,2752,88,-36,0,-1045,-2540,2169,0,-1759,
1 -241,1054,1558,-17,-2622,0,-183,587,1769,0,470,
1 2908,2693,2569,72,-431,0,-913,-2061,2107,0,-1418,
```

```
1  23,1192,1642,-10,-2439,0,-245,365,1798,0,312,
1  2441,2450,2419,59,-757,0,-805,-1668,2057,0,-1138,
1  308,1340,1734,-1,-2241,0,-311,126,1828,0,141,
1  1933,2186,2256,44,-1110,0,-687,-1241,2003,0,-833,
1  1385,1900,2080,29,-1492,0,-560,-780,1944,0,-504,
1  1805,1316,1631,332,-3861,0,-605,-822,1866,0,-227,
1  1805,1316,1631,332,-3861,0,-605,-822,1866,0,-227,
1  1736,-3116,351,-310,-5,0,-1559,-837,-1554,-2108,-2520,
1  518,-931,105,-93,-2,0,-466,-250,-464,-630,-753,
1  1646,-2954,333,-294,-5,0,-1478,-793,-1473,-1999,-2389,
1  608,-1093,123,-109,-2,0,-547,-294,-545,-740,-884,
1  1555,-2792,315,-278,-5,0,-1397,-750,-1392,-1889,-2259,
1  699,-1255,141,-125,-2,0,-628,-337,-626,-849,-1015,
1  1110,-2013,228,-189,921,0,-1013,-544,-1009,-1369,-1637,
1  1127,-2023,228,-201,-4,0,-1013,-544,-1009,-1369,-1637,
1  2441,2450,2419,59,-757,0,-805,-1668,2057,0,-1138,
1  308,1340,1734,-1,-2241,0,-311,126,1828,0,141,
1  1379,1904,2080,32,-1076,0,-560,-780,1944,0,-504,
1  1379,1904,2080,32,-1076,0,-560,-780,1944,0,-504,
1  1803,932,1305,520,455,105,-50,-1556,503,0,-1164,
1  1803,932,1305,520,455,105,-50,-1556,503,0,-1164,
1  2606,1347,1694,675,290,66,-83,-2411,542,0,-1869,
1  894,448,858,334,-137,-32,-11,-575,457,0,-356,
1  1814,925,1305,513,-273,-64,-50,-1556,503,0,-1164,
1  1814,925,1305,513,-273,-64,-50,-1556,503,0,-1164,
1  1390,1897,2080,26,-1908,0,-560,-780,1944,0,-504,
1  1390,1897,2080,26,-1908,0,-560,-780,1944,0,-504,
1  2441,2450,2419,59,-757,0,-805,-1668,2057,0,-1138,
1  308,1340,1734,-1,-2241,0,-311,126,1828,0,141,
1  1805,1316,1631,332,-3861,0,-605,-822,1866,0,-227,
1  1805,1316,1631,332,-3861,0,-605,-822,1866,0,-227,
1  1555,-2792,315,-278,-5,0,-1397,-750,-1392,-1889,-2259,
1  699,-1255,141,-125,-2,0,-628,-337,-626,-849,-1015,
1  1110,-2013,228,-189,921,0,-1013,-544,-1009,-1369,-1637,
1  1127,-2023,228,-201,-4,0,-1013,-544,-1009,-1369,-1637,
1  1555,-2792,315,-278,-5,0,-1397,-750,-1392,-1889,-2259,
1  699,-1255,141,-125,-2,0,-628,-337,-626,-849,-1015,
1  1110,-2013,228,-189,921,0,-1013,-544,-1009,-1369,-1637,
1  1127,-2023,228,-201,-4,0,-1013,-544,-1009,-1369,-1637,
1  2441,2450,2419,59,-757,0,-805,-1668,2057,0,-1138,
1  308,1340,1734,-1,-2241,0,-311,126,1828,0,141,
1  1379,1904,2080,32,-1076,0,-560,-780,1944,0,-504,
1  1379,1904,2080,32,-1076,0,-560,-780,1944,0,-504,
1  1803,932,1305,520,455,105,-50,-1556,503,0,-1164,
1  1803,932,1305,520,455,105,-50,-1556,503,0,-1164,
1  2606,1347,1694,675,290,66,-83,-2411,542,0,-1869,
1  894,448,858,334,-137,-32,-11,-575,457,0,-356,
1  1814,925,1305,513,-273,-64,-50,-1556,503,0,-1164,
1  1814,925,1305,513,-273,-64,-50,-1556,503,0,-1164,
1  1390,1897,2080,26,-1908,0,-560,-780,1944,0,-504,
1  1390,1897,2080,26,-1908,0,-560,-780,1944,0,-504,
1  2441,2450,2419,59,-757,0,-805,-1668,2057,0,-1138,
1  166,1266,1688,-6,-2340,0,-278,246,1813,0,227,
1  1805,1316,1631,332,-3861,0,-605,-822,1866,0,-227,
1  1805,1316,1631,332,-3861,0,-605,-822,1866,0,-227,
```

```
1 1555,-2792,315,-278,-5,0,-1397,-750,-1392,-1889,-2259,
1 699,-1255,141,-125,-2,0,-628,-337,-626,-849,-1015,
1 1110,-2013,228,-189,921,0,-1013,-544,-1009,-1369,-1637,
1 1127,-2023,228,-201,-4,0,-1013,-544,-1009,-1369,-1637,
1 0,0,0,0,0,0,0,0,0,0,0,
1 /
END
```
*

```
C        TITLE-AMATBL.FOR :        10-DEC-75
C
C        THIS PROGRAM SUPPLYS THE LOAD TABLES FOR THE
C        AMAVS TEST PROGRAM  AND MUST BE LINKED AS FOLLOWS:
C        AMAVS3<AMAPRX,AMAPRT,AMACAL,AMAHLD,AMAVS3,AMATBL/F/L/I
C
         DIMENSION R(281,11)
C
         DATA R/42417.,13764.,40170.,16011.,37922.,17978.,
         1 27823.,28091.,33880.,11903.,31819.,13964.,
         1 30217.,15566.,22707.,22892.,75448.,4118.,
         1 63881.,14721.,54241.,22914.,40645.,40645.,
         1 41578.,41578.,74805.,7003.,65843.,16355.,
         1 58050.,22979.,51036.,41684.,41684.,28435.,
         1 73052.,4435.,61750.,15737.,52467.,25020.,
         1 47623.,29864.,73927.,9116.,66520.,15411.,
         1 59113.,20967.,50595.,38744.,38744.,25781.,
         1 38658.,38658.,-937.,-937.,286.,-1932.,
         1 -57.,-1609.,-359.,-1370.,-612.,-858.,
         1 -2292.,-1624.,-2183.,-1714.,-2078.,-1787.,
         1 -203.,-2637.,-581.,-2282.,-912.,-2021.,
         1 -779.,-779.,50159.,50159.,78304.,14783.,
         1 73322.,22256.,65849.,28484.,60867.,50073.,
         1 50073.,34711.,74897.,12629.,67370.,19471.,
         1 60527.,25288.,52316.,40684.,40684.,30420.,
         1 74805.,7003.,65843.,16355.,58050.,22979.,
         1 51036.,41684.,41600.,41600.,443.,443.,
         1 -1040.,54.,-922.,-63.,-829.,-147.,
         1 -71.,1123.,54.,997.,155.,904.,
         1 1405.,2508.,1514.,2398.,1607.,2306.,
         1 4602.,526.,3583.,526.,2930.,-819.,
         1 2727.,-717.,2360.,-656.,1952.,-533.,
         1 -3221.,-1962.,-3096.,-2095.,-2987.,-2205.,
         1 -1574.,-3617.,-1653.,-3512.,-1784.,-3381.,
         1 -2515.,-2515.,2978.,2978.,1721.,-3053.,
         1 1195.,-2567.,709.,-2082.,5524.,265.,
         1 4958.,791.,4432.,1317.,9595.,4700.,
         1 9029.,5266.,8543.,5752.,9732.,-1805.,
         1 8806.,-1449.,7239.,-666.,2979.,2979.,
         1 34675.,34675.,64780.,6944.,56835.,12346.,
         1 49844.,13935.,64439.,8617.,57340.,13457.,
         1 50242.,18620.,63066.,5302.,55131.,10698.,
         1 48149.,12285.,34694.,34694.,26586.,26586.,
         1 69576.,-16606.,57787.,-4817.,48437.,3719.,
         1 39493.,13476.,66496.,-4591.,55619.,458.,
         1 46684.,5896.,36973.,26485.,34524.,34524.,
         1 33202.,9917.,31477.,11642.,29752.,13367.,
         1 21241.,21560.,46684.,5896.,26384.,26384.,
         1 34488.,34488.,49844.,17113.,34694.,34694.,
         1 26586.,26586.,46684.,5896.,34524.,34524.,
         1 29752.,13367.,21241.,21560.,29752.,13367.,
         1 21241.,21560.,46684.,5896.,26384.,26384.,
         1 34488.,34488.,49844.,17113.,34694.,34694.,
         1 26586.,26586.,46684.,3177.,34524.,34524.,
         1 29752.,13367.,21241.,21560.,0.,
```

```
2 -77262.,-25071.,-73168.,-29165.,-69075.,-32746.,
2 -51218.,-51167.,-75480.,-26520.,-70890.,-31110.,
2 -67320.,-34680.,-51110.,-51000.,65596.,15279.,
2 57437.,22759.,50637.,28538.,41182.,41182.,
2 20053.,20053.,35928.,3295.,31615.,7796.,
2 27864.,10984.,24488.,19987.,19987.,13610.,
2 28238.,20223.,26918.,21543.,25834.,22627.,
2 25268.,23193.,45621.,6218.,41118.,10045.,
2 36614.,13423.,31436.,24231.,24231.,16350.,
2 24285.,24285.,-1172.,-1172.,-1228.,-1212.,
2 -1225.,-1214.,-1223.,-1216.,-1221.,-1220.,
2 -2612.,-1468.,-2424.,-1622.,-2246.,-1746.,
2 -626.,-705.,-638.,-693.,-649.,-685.,
2 -1268.,-1268.,11640.,11640.,24737.,-4610.,
2 22435.,-1157.,18983.,1719.,16681.,11694.,
2 11694.,4596.,56405.,842.,49688.,6948.,
2 43582.,12137.,36255.,25876.,25876.,16717.,
2 35928.,3295.,31615.,7796.,27864.,10984.,
2 24488.,19987.,20039.,20039.,1948.,1948.,
2 -17.,651.,54.,579.,111.,528.,
2 1531.,2262.,1608.,2185.,1670.,2128.,
2 3774.,4448.,3841.,4381.,3897.,4325.,
2 8217.,1897.,6637.,1897.,5625.,-188.,
2 5309.,-30.,4741.,64.,4109.,253.,
2 -3258.,-2376.,-3170.,-2469.,-3094.,-2546.,
2 -2291.,-3348.,-2332.,-3294.,-2399.,-3226.,
2 -2869.,-2869.,9805.,9805.,8374.,4638.,
2 7962.,5018.,7582.,5398.,11913.,7798.,
2 11470.,8209.,11059.,8621.,15775.,11944.,
2 15331.,12387.,14951.,12767.,16287.,5434.,
2 15416.,5769.,13942.,6506.,9805.,9805.,
2 17470.,17470.,33167.,3193.,29050.,5993.,
2 25427.,6817.,21876.,4187.,19627.,5721.,
2 17377.,7357.,34615.,4582.,30490.,7387.,
2 26859.,8212.,17458.,17458.,35800.,35800.,
2 33761.,37962.,34335.,37388.,34791.,36971.,
2 35227.,36496.,56411.,19901.,50825.,22494.,
2 46236.,25287.,41248.,35862.,24833.,24833.,
2 -58777.,-17556.,-55723.,-20610.,-52670.,-23663.,
2 -37976.,-38167.,46236.,25287.,35924.,35924.,
2 17586.,17586.,25427.,8464.,17458.,17458.,
2 35800.,35800.,46236.,25287.,24833.,24833.,
2 -52670.,-23663.,-37976.,-38167.,-52670.,-23663.,
2 -37976.,-38167.,46236.,25287.,35924.,35924.,
2 17586.,17586.,25427.,8464.,17458.,17458.,
2 35800.,35800.,46236.,23891.,24833.,24833.,
2 -52670.,-23663.,-37976.,-38167.,0.,
3 12080.,3920.,11440.,4560.,10800.,5120.,
3 7900.,8000.,9636.,3385.,9050.,3971.,
3 8594.,4427.,6311.,6511.,32960.,6320.,
3 28640.,10280.,25040.,13340.,20000.,20000.,
3 22000.,22000.,35600.,7760.,31920.,11600.,
3 28720.,14320.,25840.,22000.,22000.,16560.,
3 43700.,6300.,37540.,12460.,32480.,17520.,
3 29840.,20160.,44000.,9000.,40000.,12400.,
```

```
3 36000.,15400.,31400.,25000.,25000.,18000.,
3 25000.,25000.,8700.,8700.,27445.,-8895.,
3 21810.,-3605.,16865.,305.,12725.,8700.,
3 10635.,-4490.,8160.,-2455.,5795.,-805.,
3 27560.,-10360.,21680.,-4840.,16520.,-760.,
3 8700.,8700.,17500.,17500.,29060.,3050.,
3 27020.,6110.,23960.,8660.,21920.,17500.,
3 17500.,11210.,38000.,5240.,34040.,8840.,
3 30440.,11900.,26120.,20000.,20000.,14600.,
3 35600.,7760.,31920.,11600.,28720.,14320.,
3 25840.,22000.,22000.,22000.,19000.,19000.,
3 29500.,3500.,26700.,6300.,24500.,8300.,
3 33200.,4800.,30200.,7800.,27800.,10000.,
3 35500.,9300.,32900.,11900.,30700.,14100.,
3 29000.,19000.,26500.,19000.,24900.,15700.,
3 24400.,15950.,23500.,16100.,22500.,16400.,
3 13500.,-18700.,10300.,-15300.,7500.,-12500.,
3 -160.,-4840.,-340.,-4600.,-640.,-4300.,
3 -2500.,-2500.,25000.,25000.,31250.,1750.,
3 28000.,4750.,25000.,7750.,41250.,8750.,
3 37750.,12000.,34500.,15250.,50200.,19950.,
3 46700.,23450.,43700.,26450.,41320.,13780.,
3 39110.,14630.,35370.,16500.,25000.,25000.,
3 16000.,16000.,25458.,7338.,22969.,9030.,
3 20778.,9528.,22550.,6980.,20570.,8330.,
3 18590.,9770.,25838.,7702.,23347.,9396.,
3 21155.,9894.,16000.,16000.,25500.,25500.,
3 38220.,12780.,34740.,16260.,31980.,18780.,
3 29340.,21660.,33740.,19100.,31500.,20140.,
3 29660.,21260.,27660.,25500.,20000.,20000.,
3 4312.,1288.,4088.,1512.,3864.,1736.,
3 2800.,2800.,29660.,21260.,25500.,25500.,
3 16000.,16000.,20778.,10524.,16000.,16000.,
3 25500.,25500.,29660.,21260.,20000.,20000.,
3 3864.,1736.,2800.,2800.,3864.,1736.,
3 2800.,2800.,29660.,21260.,25500.,25500.,
3 16000.,16000.,20778.,10524.,16000.,16000.,
3 25500.,25500.,29660.,20700.,20000.,20000.,
3 3864.,1736.,2800.,2800.,0.,
4 -21434.,-6955.,-20298.,-8091.,-19163.,-9084.,
4 -13579.,-14195.,-20165.,-7085.,-18938.,-8311.,
4 -17985.,-9265.,-12780.,-13625.,4828.,22677.,
4 7723.,20024.,10135.,17974.,13600.,13600.,
4 10103.,10103.,13981.,5843.,12905.,6965.,
4 11970.,7761.,11128.,10006.,10006.,8415.,
4 5350.,1999.,4798.,2551.,4345.,3004.,
4 4108.,3241.,3372.,3929.,3436.,3875.,
4 3500.,3827.,3573.,3675.,3675.,3786.,
4 3754.,3754.,49857.,49857.,116694.,-13036.,
4 96577.,5848.,78924.,19806.,64144.,49776.,
4 94782.,8817.,80715.,20383.,67273.,29761.,
4 116505.,-11091.,96720.,7482.,79357.,21211.,
4 49695.,49695.,15810.,15810.,19435.,11456.,
4 18809.,12394.,17870.,13177.,17244.,15889.,
4 15889.,13959.,16379.,9191.,15510.,9981.,
4 14720.,10652.,13772.,12430.,12430.,11245.,
```

```
4 13981.,5843.,12905.,6965.,11970.,7761.,
4 11128.,10006.,10082.,10082.,26685.,26685.,
4 21066.,-1947.,18588.,530.,16640.,2301.,
4 39168.,14029.,36512.,16685.,34388.,18632.,
4 61961.,38770.,59659.,41071.,57712.,43019.,
4 94757.,26599.,77717.,26599.,66812.,4106.,
4 63404.,5810.,57270.,6833.,50454.,8877.,
4 68574.,54148.,67140.,55671.,65886.,56926.,
4 75063.,47748.,74013.,49149.,72262.,50900.,
4 61322.,61322.,14964.,14964.,8935.,4800.,
4 8480.,5220.,8059.,5641.,17326.,12769.,
4 16835.,13225.,16379.,13681.,26967.,22725.,
4 26476.,23216.,26055.,23637.,30753.,4250.,
4 28626.,5068.,25027.,6868.,14962.,14962.,
4 14651.,14651.,23701.,6508.,21339.,8114.,
4 19261.,8586.,20668.,5904.,18790.,7184.,
4 16913.,8550.,23736.,6546.,21375.,8152.,
4 19297.,8624.,14630.,14630.,747.,747.,
4 -2109.,3789.,-1302.,2982.,-662.,2398.,
4 -50.,1730.,2528.,-471.,2069.,-258.,
4 1692.,-28.,1282.,840.,9472.,9472.,
4 -8819.,-2634.,-8361.,-3092.,-7903.,-3550.,
4 -5374.,-5727.,1692.,-28.,933.,933.,
4 14824.,14824.,19261.,9531.,14630.,14630.,
4 747.,747.,1692.,-28.,9472.,9472.,
4 -7903.,-3550.,-5374.,-5727.,-7903.,-3550.,
4 -5374.,-5727.,1692.,-28.,933.,933.,
4 14824.,14824.,19261.,9531.,14630.,14630.,
4 747.,747.,1692.,-143.,9472.,9472.,
4 -7903.,-3550.,-5374.,-5727.,0.,
8 -472.,-153.,-447.,-178.,-422.,-200.,
8 72546.,-313.,-414.,-145.,-389.,-170.,
8 -369.,-190.,82962.,-280.,183782.,-125737.,
8 133589.,-79727.,91762.,-44174.,60297.,60297.,
8 42669.,42669.,51685.,-15276.,42834.,-6040.,
8 35137.,501.,28210.,18974.,18974.,5889.,
8 89828.,-75752.,62556.,-48480.,40154.,-26078.,
8 28466.,-14390.,26901.,-9689.,22719.,-6134.,
8 18537.,-2998.,13728.,7038.,7038.,-280.,
8 25924.,25924.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
8 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
8 0.,0.,-15450.,-15450.,15197.,-11265.,
8 13121.,-8152.,10008.,-5557.,7932.,3436.,
8 3436.,-2963.,8547.,-11853.,6081.,-9611.,
8 3839.,-7706.,1149.,-2662.,-2662.,-6024.,
8 51685.,-15276.,42834.,-6040.,35137.,501.,
8 28210.,18974.,37595.,37595.,0.,0.,
8 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
8 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
8 0.,0.,0.,0.,-190.,-190.,22186.,-19274.,
8 17619.,-15058.,13402.,-10841.,29855.,-15823.,
8 24935.,-11255.,20368.,-6687.,34770.,-7745.,
8 29851.,-2826.,25634.,1390.,17691.,-323.,
8 16245.,232.,13799.,1456.,-280.,-280.,
8 -13077.,-13077.,29351.,-16259.,23086.,-11998.,
8 17573.,-10745.,35512.,-8306.,29940.,-4507.,
```

```
8 24367.,-454.,-214514.,-247430.,-219035.,-244356.,
8 -223014.,-243451.,-16442.,-16442.,-115232.,-115232.,
8 32197.,-212385.,-1259.,-178928.,-27794.,-154700.,
8 -53175.,-127012.,-2123.,-158420.,-26038.,-147317.,
8 -45681.,-135360.,-67033.,-90094.,-233281.,-233281.,
8 -294.,-87.,-278.,-103.,-263.,-118.,
8 55692.,-191.,-45681.,-135360.,-64956.,-64956.,
8 27530.,27530.,17573.,-8239.,-16442.,-16442.,
8 -115232.,-115232.,-45681.,-135360.,-233281.,-233281.,
8 -263.,-118.,55692.,-191.,-263.,-118.,
8 55692.,-191.,-45681.,-135360.,-64956.,-64956.,
8 27530.,27530.,17573.,-8239.,-16442.,-16442.,
8 -115232.,-115232.,-45681.,-141338.,-233281.,-233281.,
8 -263.,-118.,55692.,-191.,0.,
9 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
9 0.,0.,0.,0.,5420.,5420.,6500.,-1921.,5387.,-759.,
9 4419.,63.,3548.,2386.,2386.,740.,
9 11297.,-9527.,7867.,-6097.,5049.,-3279.,
9 3579.,-1809.,3383.,-1218.,2857.,-771.,
9 2331.,-377.,1726.,885.,885.,-35.,
9 3303.,3303.,-6333.,-6333.,-32161.,-17902.,
9 -29950.,-19978.,-28010.,-21512.,-26385.,-24806.,
9 -39626.,-27865.,-37701.,-29448.,-35862.,-30730.,
9 -44032.,-131082.,-57530.,-118410.,-69376.,-109044.,
9 -43279.,-43279.,-1986.,-1986.,1911.,-1416.,
9 1650.,-1025.,1258.,-699.,997.,432.,
9 432.,-372.,1075.,-1490.,764.,-1208.,
9 483.,-968.,144.,-334.,-334.,-757.,
9 6500.,-1921.,5387.,-759.,4419.,63.,
9 3548.,2386.,4771.,4771.,73135.,73135.,
9 82913.,-4613.,73487.,4812.,66081.,11545.,
9 102723.,7117.,92624.,17216.,84545.,24622.,
9 120762.,32562.,112010.,41315.,104603.,48721.,
9 118002.,54920.,102232.,54920.,92138.,34103.,
9 88984.,35680.,83307.,36626.,76999.,38519.,
9 135957.,-47057.,117769.,-27732.,101855.,-11818.,
9 70152.,19884.,68218.,22462.,64996.,25684.,
9 27028.,27028.,-309.,-309.,36053.,-31321.,
9 28630.,-24469.,21779.,-17618.,48514.,-25712.,
9 40520.,-18289.,33098.,-10867.,56501.,-12586.,
9 48507.,-4592.,41656.,2258.,28748.,-525.,
9 26399.,378.,22423.,2366.,-455.,-455.,
9 -1687.,-1687.,3691.,-2044.,2903.,-1508.,
9 2210.,-1350.,4466.,-1044.,3765.,-566.,
9 3064.,-57.,-26978.,-31117.,-27546.,-30731.,
9 -28047.,-30617.,-2118.,-2118.,0.,0.,
9 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
9 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
9 3513.,3513.,2210.,-1035.,-2118.,-2118.,
9 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
9 3513.,3513.,2210.,-1035.,-2118.,-2118.,
9 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
2 -90581.,-29393.,-85782.,-34192.,-80983.,-38392.,
2 -59987.,-59987.,-129699.,-45569.,-121811.,-53457.,
2 -115677.,-59591.,-87634.,-87634.,-42388.,-10384.,
2 -37198.,-15142.,-32873.,-18818.,-26819.,-26819.,
2 -7248.,-7248.,-13821.,-366.,-12042.,-2222.,
```

```
2 -10469.,-3536.,-9104.,-7248.,-7248.,-4619.,
2 22479.,-37080.,12669.,-27270.,4611.,-19212.,
2 407.,-15008.,-14014.,-1646.,-12601.,-2847.,
2 -11187.,-3908.,-9562.,-7300.,-7300.,-4826.,
2 -7300.,-7300.,18071.,18071.,37846.,-490.,
2 31901.,5089.,26684.,9214.,22317.,18071.,
2 35380.,3660.,30190.,7928.,25230.,11388.,
2 37665.,-2139.,31493.,3654.,26076.,7937.,
2 18071.,18071.,-6035.,-6035.,-10012.,-1063.,
2 -9310.,-2116.,-8258.,-2994.,-7556.,-6035.,
2 -6035.,-3871.,-28850.,-2709.,-25690.,-5581.,
2 -22817.,-8023.,-19370.,-14487.,-14487.,-10178.,
2 -13821.,-366.,-12042.,-2222.,-10496.,-3536.,
2 -9104.,-7248.,-7248.,-7248.,32445.,32445.,
2 70962.,-20048.,61161.,-10247.,53460.,-3246.,
2 82151.,-17260.,71650.,-6759.,63249.,941.,
2 87735.,-3975.,78634.,5125.,70933.,12826.,
2 60400.,32445.,53411.,32445.,48938.,23220.,
2 47541.,23919.,45025.,24338.,42229.,25177.,
2 73724.,-20841.,64326.,-10856.,56103.,-2633.,
2 29905.,23564.,29661.,23889.,29254.,24296.,
2 26735.,26735.,25604.,25604.,39199.,-9782.,
2 33803.,-4800.,28822.,180.,52586.,-1377.,
2 46774.,4019.,41378.,9415.,63585.,13358.,
2 57774.,19169.,52792.,24150.,46524.,11222.,
2 43691.,12311.,38897.,14708.,25604.,25604.,
2 -1553.,-1553.,-3681.,395.,-3121.,14.,
2 -2628.,-97.,13215.,1355.,11707.,2384.,
2 10199.,3480.,-6691.,-2486.,-6114.,-2879.,
2 -5606.,-2995.,-1553.,-1553.,-17725.,-17725.,
2 306.,-35757.,-4626.,-30824.,-8539.,-27251.,
2 -12281.,-23169.,-33090.,-5791.,-28913.,-7731.,
2 -25482.,-9819.,-21753.,-17725.,-19156.,-19156.,
2 -49370.,-14747.,-46806.,-17311.,-44241.,-19876.,
2 -32059.,-32059.,-25482.,-9819.,-17725.,-17725.,
2 -1553.,-1553.,-2628.,-321.,-1553.,-1553.,
2 -17725.,-17725.,-25482.,-9819.,-19156.,-19156.,
2 -44241.,-19876.,-32059.,-32059.,-44241.,-19876.,
2 -32059.,-32059.,-25482.,-9819.,-17725.,-17725.,
2 -1553.,-1553.,-2628.,-321.,-1553.,-1553.,
2 -17725.,-17725.,-25482.,-8775.,-19156.,-19156.,
2 -44241.,-19876.,-32059.,-32059.,0.,
3 -9845.,-3194.,-9323.,-3716.,-8802.,-4172.,
3 -6520.,-6520.,-9899.,-3478.,-9297.,-4080.,
3 -8829.,-4548.,-6689.,-6689.,-11563.,-1129.,
3 -9871.,-2680.,-8461.,-3878.,-6487.,-6487.,
3 -6421.,-6421.,-12705.,159.,-11005.,-1614.,
3 -9526.,-2871.,-8195.,-6421.,-6421.,-3907.,
3 -6790.,-6809.,-6793.,-6806.,-6796.,-6803.,
3 -6797.,-6802.,-13969.,-762.,-12460.,-2045.,
3 -10950.,-3177.,-9215.,-6800.,-6800.,-4158.,
3 -6800.,-6800.,-6108.,-6108.,-16716.,3849.,
3 -13527.,855.,-10728.,-1357.,-8385.,-6108.,
3 -9818.,769.,-8086.,-655.,-6430.,-1810.,
3 -16752.,3531.,-13607.,578.,-10847.,-1604.,
3 -6108.,-6108.,-2254.,-2254.,-4262.,256.,
3 -3908.,-274.,-3376.,-717.,-3022.,-2254.,
```

```
3 -2254.,-1161.,-13860.,-537.,-12249.,-2001.,
3 -10785.,-3246.,-9028.,-6540.,-6540.,-4344.,
3 -12705.,159.,-11005.,-1614.,-9526.,-2871.,
3 -8195.,-6421.,-6421.,-6421.,-3880.,-3880.,
3 2098.,-2681.,1583.,-2166.,1179.,-1799.,
3 -1269.,-6490.,-1820.,-5939.,-2262.,-5534.,
3 -6513.,-11330.,-6991.,-10852.,-7396.,-10448.,
3 -18234.,-3880.,-14645.,-3880.,-12348.,856.,
3 -11631.,497.,-10339.,282.,-8903.,-147.,
3 -4290.,4887.,-3378.,3918.,-2580.,3120.,
3 -1405.,1945.,-1276.,1773.,-1061.,1558.,
3 270.,270.,-1761.,-1761.,861.,-288.,
3 735.,-171.,618.,-54.,-1127.,-2394.,
3 -1263.,-2268.,-1390.,-2141.,-3633.,-4812.,
3 -3769.,-4676.,-3886.,-4559.,-5692.,941.,
3 -5159.,736.,-4258.,286.,-1761.,-1761.,
3 -6925.,-6925.,-14463.,-21.,-12479.,-1370.,
3 -10733.,-1767.,-7639.,-177.,-6690.,-824.,
3 -5741.,-1514.,-14678.,-205.,-12690.,-1557.,
3 -10940.,-1955.,-6925.,-6925.,-3469.,-3469.,
3 -3208.,-3729.,-3279.,-3658.,-3336.,-3606.,
3 -3390.,-3547.,-11306.,2618.,-9175.,1629.,
3 -7425.,563.,-5523.,-3469.,-3656.,-3656.,
3 -3725.,-1112.,-3531.,-1306.,-3338.,-1499.,
3 -2419.,-2419.,-7425.,563.,-3469.,-3469.,
3 -6925.,-6925.,-10733.,-2560.,-6925.,-6925.,
3 -3469.,-3469.,-7425.,563.,-3656.,-3656.,
3 -3338.,-1499.,-2419.,-2419.,-3338.,-1499.,
3 -2419.,-2419.,-7425.,563.,-3469.,-3469.,
3 -6925.,-6925.,-10733.,-2560.,-6925.,-6925.,
3 -3469.,-3469.,-7425.,1096.,-3656.,-3656.,
3 -3338.,-1499.,-2419.,-2419.,0.,
4 2183.,708.,2067.,824.,1952.,925.,
4 1446.,1446.,1916.,673.,1800.,789.,
4 1709.,880.,1295.,1295.,2173.,737.,
4 1940.,951.,1746.,1116.,1475.,1475.,
4 1535.,1535.,2097.,945.,1945.,1104.,
4 1813.,1217.,1693.,1535.,1535.,1309.,
4 277.,2110.,579.,1808.,827.,1560.,
4 956.,1431.,1692.,774.,1587.,863.,
4 1482.,942.,1362.,1194.,1194.,1010.,
4 1194.,1194.,1816.,1816.,4189.,-411.,
4 3475.,258.,2849.,753.,2325.,1816.,
4 -8813.,690.,-7258.,-588.,-5772.,-1625.,
4 4157.,-699.,3404.,7.,2743.,529.,
4 1816.,1816.,-2022.,-2022.,-3825.,232.,
4 -3507.,-245.,-3029.,-642.,-2711.,-2022.,
4 -2022.,-1040.,2156.,831.,1995.,976.,
4 1850.,1100.,1675.,1428.,1428.,1209.,
4 2097.,945.,1945.,1104.,1813.,1217.,
4 1693.,1535.,1535.,1535.,3816.,3816.,
4 2401.,4373.,2613.,4161.,2780.,4009.,
4 2738.,4893.,2966.,4665.,3148.,4498.,
4 3429.,5417.,3627.,5219.,3793.,5053.,
4 5530.,3816.,5101.,3816.,4827.,3250.,
4 4741.,3293.,4587.,3318.,4415.,3370.,
4 1986.,-1524.,1637.,-1153.,1332.,-848.,
```

```
4 -1261.,1745.,-1146.,1591.,-953.,1398.,
4 242.,242.,-1581.,-1581.,772.,-259.,
4 659.,-154.,554.,-49.,-1012.,-2149.,
4 -1134.,-2036.,-1248.,-1922.,-3261.,-4319.,
4 -3383.,-4197.,-3488.,-4092.,-5109.,844.,
4 -4631.,660.,-3822.,256.,-1581.,-1581.,
4 1083.,1083.,1251.,929.,1206.,959.,
4 1167.,967.,-6857.,-160.,-6005.,-741.,
4 -5154.,-1360.,1056.,763.,1016.,791.,
4 981.,799.,1083.,1083.,4185.,4185.,
4 3472.,4897.,3667.,4702.,3822.,4561.,
4 3969.,4400.,4668.,3809.,4536.,3870.,
4 4428.,3936.,4311.,4185.,4016.,4016.,
4 -3343.,-998.,-3169.,-1172.,-2995.,-1346.,
4 -2171.,-2171.,4428.,3936.,4185.,4185.,
4 1083.,1083.,1167.,985.,1083.,1083.,
4 4185.,4185.,4428.,3936.,4016.,4016.,
4 -2995.,-1346.,-2171.,-2171.,-2995.,-1346.,
4 -2171.,-2171.,4428.,3936.,4185.,4185.,
4 1083.,1083.,1167.,985.,1083.,1083.,
4 4185.,4185.,4428.,3903.,4016.,4016.,
4 -2995.,-1346.,-2171.,-2171.,0.,
5 -70452.,-22862.,-66719.,-26594.,-62987.,-29860.,
5 -46657.,-46657.,-89063.,-31292.,-83647.,-36708.,
5 -79435.,-40921.,-60178.,-60178.,
5 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
5 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
5 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
5 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
5 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
5 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
5 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
5 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
5 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
5 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
5 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
5 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
5 -45842.,-13693.,-43461.,-16074.,-41079.,-18456.,
5 -29768.,-29768.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
5 0.,0.,0.,0.,-41079.,-18456.,-29768.,-29768.,
5 -41079.,-18456.,-29768.,-29768.,
5 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
5 -41079.,-18456.,-29768.,-29768.,0.,
6 -43624.,-14156.,-41313.,-16467.,-39002.,-18489.,
6 -28890.,-28890.,-59205.,-20801.,-55604.,-24402.,
6 -52804.,-27202.,-40003.,-40003.,-20252.,11747.,
6 -15063.,6990.,-10739.,3315.,-4685.,-4685.,
6 -15887.,-15887.,-33352.,2400.,-28626.,-2530.,
6 -24517.,-6023.,-20818.,-15887.,-15887.,-8900.,
6 -49273.,14930.,-38698.,4355.,-30012.,-4330.,
6 -25480.,-8862.,-36851.,-599.,-32708.,-4120.,
6 -28565.,-7228.,-23800.,-17171.,-17171.,-9921.,
6 -17171.,-17171.,-11792.,-11792.,-32317.,7474.,
6 -26147.,1681.,-20732.,-2599.,-16199.,-11792.,
6 1896.,1950.,1905.,1943.,1914.,1937.,
6 -32092.,9516.,-25640.,3459.,-19978.,-1017.,
6 -11792.,-11792.,-16508.,-16508.,-30169.,568.,
```

```
     6 -27758.,-3047.,-24142.,-6061.,-21731.,-16508.,
     6 -16508.,-9074.,-43958.,-746.,-38735.,-5494.,
     6 -33986.,-9531.,-28288.,-20215.,-20215.,-13092.,
     6 -33352.,2400.,-28626.,-2530.,-24517.,-6023.,
     6 -20818.,-15887.,-15887.,-15887.,-24607.,-24607.,
     6 -39919.,7380.,-34825.,2286.,-30823.,-1351.,
     6 -50440.,1226.,-44982.,-4231.,-40616.,-8233.,
     6 -59929.,-12265.,-55199.,-16995.,-51197.,-20997.,
     6 -57957.,-24607.,-49619.,-24607.,-44283.,-13601.,
     6 -42616.,-14435.,-39614.,-14935.,-36279.,-15936.,
     6 -53309.,-11909.,-49195.,-16280.,-45595.,-19880.,
     6 -40778.,-24697.,-40159.,-25522.,-39128.,-26553.,
     6 -32738.,-32738.,-11362.,-11362.,-13564.,5962.,
     6 -11413.,3977.,-9427.,1991.,-22119.,-605.,
     6 -19802.,-2757.,-17651.,-4908.,-30365.,-10341.,
     6 -28048.,-12658.,-26062.,-14644.,-25880.,-1381.,
     6 -23914.,-2137.,-20587.,-3801.,-11362.,-11362.,
     6 -17029.,-17029.,-37452.,1674.,-32077.,-1980.,
     6 -27348.,-3055.,-29428.,-378.,-25734.,-2897.,
     6 -22040.,-5584.,-29673.,9039.,-24355.,5423.,
     6 -19676.,4360.,-17029.,-17029.,-7372.,-7372.,
     6 -46976.,32232.,-36141.,21397.,-27547.,13551.,
     6 -19328.,4584.,-25733.,6889.,-20741.,4571.,
     6 -16641.,2076.,-12185.,-7372.,-3318.,-3318.,
     6 -36878.,-11015.,-34962.,-12931.,-33046.,-14847.,
     6 -23947.,-23947.,-16641.,2076.,-7372.,-7372.,
     6 -17029.,-17029.,-27348.,-5205.,-17029.,-17029.,
     6 -7372.,-7372.,-16641.,2076.,-3318.,-3318.,
     6 -33046.,-14847.,-23947.,-23947.,-33046.,-14847.,
     6 -23947.,-23947.,-16641.,2076.,-7372.,-7372.,
     6 -17029.,-17029.,-27348.,-5205.,-17029.,-17029.,
     6 -7372.,-7372.,-16641.,3323.,-3318.,-3318.,
     6 -33046.,-14847.,-23947.,-23947.,0.,/
C
      CALL AMAVS(R)
      STOP' END OF RUN'
      END
?*
```

```
  TT:<AMAVS3.FOR
 IS PROPORTIONAL TO NPONTS/300.

C       STAND ALONE AMAVS FUNCTION GENERATOR.
C       THIS PROGRAM MUST BE LINKED AS FOLLOWS:
C       AMAVS<AMAPRX,AMAHLD,AMAVS,AMATBL/F/L/I
C
C       PARAMETERS ARE: HAVSIN RATE       =INCTM
C                       HOLD TIME         =IHLDTM
C                       FLITE NUMBER      =M
C                       LAST FLITE        =MSTOP
C                       START CYCLE NO.   =INTCYL
C
C
        SUBROUTINE AMAVS(R)
        INTEGER CYCLE,REF
        DIMENSION R(281,11),IOUT(12),IC(11),DIFRA(11),REF(11)
        DIMENSION CYCLE(281),PEAKLD(11),IDATO(4)
        DIMENSION OLDREF(11),IDUM(11),REFNEW(11),DIFR(11)
        DIMENSION HAVSIN(181)
C
C
C
        DATA PEAKLD/78304.,77262.,50200.,116694.,247430.,
        1 135957.,129699.,18234.,8813.,
        1 89063.,59929./
C
C
        DATA CYCLE/-2,-2,-1,-1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,
        1 -2,-2,-1,-1,1,1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,1,1,
        1 -2,-2,-1,-1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,1,1,
        1 1,1,1,1,-2,-2,-1,-1,1,1,1,1,-2,-2,-1,-1,1,1,
        1 -2,-2,-1,-1,1,1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,1,1,
        1 -2,-2,-1,-1,1,1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,1,1,1,1,
        1 -2,-2,-1,-1,1,1,-2,-2,-1,-1,1,1,-2,-2,-1,-1,
        1 1,1,1,1,1,1,1,1,1,21,1,57,1,1,-2,-2,-1,-1,
        1 1,1,-2,-2,-1,-1,1,1,1,1,1,1,-2,-2,-1,-1,1,1,-2,-2,
        1 -1,-1,1,1,-2,-2,-1,-1,1,1,1,1,1,11,1,15,1,1,1,1,
        1 -2,-2,-1,-1,1,1,-2,-2,-1,-1,1,1,-2,-2,-1,-1,
        1 1,1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,-2,-2,-1,-1,
        1 1,1,1,1,1,1,-2,-2,-1,-1,1,1,1,1,1,1,1,1,1,1,1,1,
        1 1,1,1,1,1,1,1,1,1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1,
        1 -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,1/
C
C
C       INITIALIZE
1       DO 100 J=1,11
        OLDREF(J)=0
        IOUT(J)=0
100     CONTINUE
        IOUT(12)=600
        IDATO(1)=0
        I5=0
        I6=1
        ANGLE=0.
        DO 110 J=1,181
```

```
        HAVSIN(J)=(1.-COS(ANGLE*3.1416/180.))/2.
        ANGLE=ANGLE+1.
110     CONTINUE
D       GOTO 1
        CALL JD2A(IOUT)                         ! SET OUTPUT TO 0
D       CALL JBERNY(IDATO)
C
C       SET UP TEST PARAMETERS
C
        TYPE 2
2       FORMAT (' AMAVS TEST PROGRAM',/,' ENTER TODAY"S DATE')
        ACCEPT 3
3       FORMAT (' ENTER TODAY"S DATE  ')
54      TYPE 56
56      FORMAT(' BYPASS CHECK MINI ?',/,
        1' YES=2:        NO=1',/)
        ACCEPT 57,NOCHEK
57      FORMAT(I2)
        IF(NOCHEK.LT.1.OR.NOCHEK.GT.2)GO TO 54
15      TYPE 16
16      FORMAT (' DO YOU WISH TO ENTER TEST PARAMETERS ???',/,
        1' YES = 2 ,NO=1',/)
        ACCEPT 17,IP
17      FORMAT(I3)
        IF(IP.GT.2.OR.IP.LT.1) GOTO 15
        IF(IP.EQ.1) GOTO 70
20      TYPE 21
21      FORMAT (' STARTING FLIGHT NUMBER ?',/)
        ACCEPT 22, M
22      FORMAT (I5)
        IF (M.LT.1.OR.M.GT.1280) GOTO 20
        M1=M
25      TYPE 26
26      FORMAT (' STARTING SPECTRUM STEP NUMBER ?',/,
        1'  NUMBER BETWEEN "1 & 140"',/)
        ACCEPT 27, ISTP
27      FORMAT (I5)
        IF (ISTP.LT.1.OR.ISTP.GT.140) GOTO 25
        IF(ISTP.GT.1)ISTP=(ISTP*2)-1
        ISTEP=ISTP
30      TYPE 31
31      FORMAT (' STARTING CYCLE NUMBER ?',/,
        1'  BETWEEN "1 & 29".',/)
        ACCEPT 32, ICYL
32      FORMAT (I5)
        IF (ICYL.LT.1.OR.ICYL.GT.29) GOTO 30
35      TYPE 36
36      FORMAT(' HAVERSINE  RATE ?',/,
        1' NUMBER FROM "1 TO 300" ',/)
        ACCEPT 37, INCTIM
37      FORMAT(I5)
        IF (INCTIM.LT.1.OR.INCTIM.GT.300) GOTO 35
40      TYPE 41
41      FORMAT('  HOLD TIME ?',/)
        ACCEPT 42, IHLDTM
42      FORMAT (I6)
        IF (IHLDTM.LT.1.OR.IHLDTM.GT.32000) GOTO 40
4
```

```
 5      TYPE 46
46      FORMAT('  NUMBER OF FLIGHTS TO BE RUN ?',/)
        ACCEPT 47, MSTOP
47      FORMAT(I5)
        MSTOP=(MSTOP+M)-1
        IF(MSTOP.LT.M.OR.MSTOP.GT.1280)GOTO 45
50      TYPE 51
51      FORMAT(' DO YOU WISH TO REENTER PARAMETERS ?',/,
        1' YES = 2 AND NO=1',/)
        ACCEPT 52,IDO
52      FORMAT(I2)
        IF(IDO.GT.2.OR.IDO.LT.1) GOTO 50
        IF(IDO.EQ.2) GOTO 20
70      IF(NOCHEK.EQ.1)CALL JBERNY(IOUT)
        PAUSE' TYPE CARRIAGE RETURN(CR) TO START'
C
C       THE PROGRAM WILL HOLD AT THIS POINT UNTIL A CARRIAGE RETURN
C       IS TYPED ON THE KEYBOARD
C
C
140     IOUT(12)=1500
        CALL JD2A(IOUT)                      !PRESSURIZE
        CALL ICLK(3,15000)                               !HOLD FOR PRESS.
        TYPE 90, M
90      FORMAT(/,' STARTING FLIGHT ',I5,/)
142     IF(ICLK(3).NE.0) GOTO 142
C
150     DO 700 I=ISTEP,281                               !START FUNC. GEN.
        I2=I
        MESAG4=0
        IF(I.EQ.281)I2=0
        IF(I.EQ.257.AND.(M/10*10-M).NE.0)I2=0
        IDATO(1)= I2
152     INTCYL=0
        IF(M.GT.M1)GOTO 160
        IF(I.EQ.ISTEP+1.AND.ICYL.GT.1)INTCYL=(ICYL-1)*2
160     NCYCLE=(CYCLE(I)-INTCYL)                         !SET CYCLE NO.
        IF(I.EQ.257.AND.(M/10*10-M).NE.0)NCYCLE=1
        IF(NCYCLE.LT.-2)GOTO 175
        IF(NCYCLE)170,175,180
170     IF(CYCLE(I).EQ.-1.AND.(M/10*10-M).EQ.0)GO TO 180
        IF(CYCLE(I).EQ.-2.AND.(M/100*100-M).EQ.0)GO TO 180
        GO TO 700
175     PAUSE' WRONG CYCLE NO.- REENTER PARAMETERS'      !HOLD
C
180     IF(NOCHEK.NE.2)CALL JBERNY(IDATO)
        DO 400 J=1,11               !CALC. REF.VAL. FOR EACH LOAD
        IF(R(I,J))200,190,200
190     REFNEW(J)= 0
        GO TO 300
200     REFNEW(J)=(R(I,J)/PEAKLD(J))*2047.
        IF(I.EQ.281)REFNEW(J)=0.
300     IF(I.EQ.257.AND.(M/10*10-M).NE.0)REFNEW(J)=0.
        REF(J)=ABS(REFNEW(J)-OLDREF(J))
400     CONTINUE
401     NPONTS=MAX0(REF(1),REF(2),REF(3),REF(4),
```

```
        1 REF(5),REF(6),REF(7),REF(8),
        1 REF(9),REF(10),REF(11))
        IF(NPONTS.EQ.0)NPONTS=10
        INCTM=FLOAT(INCTIM)*(FLOAT(NPONTS)/2000.)+1.
D       TYPE 99,NPONTS,INCTM
D99     FORMAT(' NPONTS =',I5,'      INCTM       =',I5)
402     DO 405 J=1,11
305     DIFR(J)=(REFNEW(J)-OLDREF(J))
405     CONTINUE
C
C
450     DO 600 K=1,181
        DO 500 J=1,11
        IOUT(J)=OLDREF(J)+DIFR(J)*HAVSIN(K)
500     CONTINUE
D       GOTO 600
        JCYCLE=(CYCLE(I)-NCYCLE)/2+1
510     CALL ICLK(1,INCTM)                              !SET RAMP RAT)E
512     IF(ICLK(1).NE.0)GO TO 512
520     CALL JD2A(IOUT)                          !OUTPUT NEW INC.  VALUE
C
        CALL JA2D(IC)
        IF(IC(1).LT.700)CALL AMAHLD(I2,I,M,JCYCLE,IC,MSTOP,
        1OLDREF,IOUT)
C
600     CONTINUE
D       GOTO 640
        IF(NCYCLE.GT.1)GOTO 630
610     DO 620 J=1,11
        OLDREF(J)=REFNEW(J)
        IOUT(J)=REFNEW(J)
620     CONTINUE
        CALL JD2A(IOUT)
630     CALL ICLK(2,IHLDTM)                             !PLATEAU TIME
        IF(IPRT.EQ.2)GOTO 632
        TYPE 80,M,I2,JCYCLE
80      FORMAT (' FLT',I5,'  LVL',I4,'  CYL',I3)
        IF(MESAG4.EQ.1)GOTO 632
        IDATO(1)=I2+16384
        IF(NOCHEK.EQ.1)CALL JBERNY(IDATO)
        MESAG4=1
632     CALL JA2D(IC)
        IF(IC(1).LT.700)CALL AMAHLD(I2,I,M,JCYCLE,IC,MSTOP,
        1OLDREF,IOUT)
638     IF(ICLK(2).NE.0)GO TO 632
C
        IF(I.EQ.257.AND.(M/10*10-M).NE.0)GOTO 710
640     NCYCLE=NCYCLE-1                          !DEC.  CYCLE NO.
        IF(NCYCLE)700,700,650
650     DO 660 J=1,11
        OLDREF(J)=OLDREF(J)+DIFR(J)
        DIFR(J)=-DIFR(J)                                !INVERT RAMP
660     CONTINUE
        GO TO 450
700     CONTINUE                                        !END OF FLIGHT ?
D
```

```
          GOTO 720
710       IOUT(12)=600                           !DEPRESSURIZE
          CALL JD2A(IOUT)
          CALL ICLK(3,8000)
712       IF(ICLK(3).NE.0) GOTO 712
          CALL JA2D(IC)
          IF(IC(1).LT.700)CALL AMAHLD(I2,I,M,CYCLE,NCYCLE,IC,STOP,
          1OLDREF,IOUT)
720       IF(M.NE.MSTOP) GOTO 725                !LAST FLIGHT ?
722       IOUT (12)= 0
D         GOTO 735
          CALL JD2A(IOUT)
          RETURN
C
725       M=M+1                                  !ADD 1 TO FLIGHT COUNT
          ISTEP=1                                !RESET STEP NO.
D         GOTO 150
D735      RETURN
          GO TO 140
          END
*
```

```
 TT:<S.FOR
C       S.FOR PROGRAM FOR AMAVS 1976 VERSION 3 8-DEC-76
        BLOCK DATA
        COMMON IN(2),IOUT(2)
        COMMON IR(12),IC(12),IO(12),RE(12)
        COMMON ALPHA(12),BETA(12),TEMPX(12)
        COMMON ISTAT(12),ITICKS
        DATA IN/2*0/,IOUT/2*0/
        DATA IR/12*0/,IC/12*0/,IO/12*0/,RE/12*0./
        DATA ALPHA/12*0./,BETA/12*0./,TEMPX/12*0./
        DATA ISTAT/6*0,100,5*0/,ITICKS/100/
        END

C       BEGIN MAINLINE OF DDC
        COMMON IN(2),IOUT(2)
        COMMON IR(12),IC(12),IO(12),RE(12)
        COMMON ALPHA(12),BETA(12),TEMPX(12)
        COMMON ISTAT(12),ITICKS
C
C       BEGIN EXECUTABLE CODE
C
C       START CLOCK
        CALL ICLK0(ITICKS)
C
C       START PRIMARY LOOP
10      CONTINUE
C
C       OPTIONAL DEBUG CODE FOR TIMING PURPOSES
C       USE SCOPE TO DETERMINE ITTERATION RATE & RATIO
C       OF WAIT TO COMPUTATION TIME
D               CALL ISD2A(0,13)
C
C       SYNC ON TIME BASE
        ISTAT(1)=IWAIT()
C
C       SECOND PART FOR TIMING
D               CALL ISD2A(4000,13)
C
C
C
C       INSERT DDC CODE AFTER HERE
C
        IF(IRTP(1,IC(1)).LT.0)ISTAT(2)=ISTAT(2).OR.2**1
        RE(1)=IR(1)-IC(1)
        IO(1)=TEMPX(1)+ALPHA(1)*RE(1)
        IF(IABS(IO(1)).GT.8000)IO(1)=ISIGN(8000,IO(1))
        IF(ISD2A(IO(1),1).LT.0)ISTAT(3)=ISTAT(3).OR.2**1
C
        IF(IRTP(2,IC(2)).LT.0)ISTAT(2)=ISTAT(2).OR.2**2
        RE(2)=IR(2)-IC(2)
        IO(2)=TEMPX(2)+ALPHA(2)*RE(2)
        IF(IABS(IO(2)).GT.8000)IO(2)=ISIGN(8000,IO(2))
        IF(ISD2A(IO(2),2).LT.0)ISTAT(3)=ISTAT(3).OR.2**2
C
        IF(IRTP(3,IC(3)).LT.0)ISTAT(2)=ISTAT(2).OR.2**3
```

```
      RE(3)=IR(3)-IC(3)
      IO(3)=TEMPX(3)+ALPHA(3)*RE(3)
      IF(IABS(IO(3)).GT.8000)IO(3)=ISIGN(8000,IO(3))
      IF(ISD2A(IO(3),3).LT.0)ISTAT(3)=ISTAT(3).OR.2**3
C
      IF(IRTP(4,IC(4)).LT.0)ISTAT(2)=ISTAT(2).OR.2**4
      RE(4)=IR(4)-IC(4)
      IO(4)=TEMPX(4)+ALPHA(4)*RE(4)
      IF(IABS(IO(4)).GT.8000)IO(4)=ISIGN(8000,IO(4))
      IF(ISD2A(IO(4),4).LT.0)ISTAT(3)=ISTAT(3).OR.2**4
C
      IF(IRTP(5,IC(5)).LT.0)ISTAT(2)=ISTAT(2).OR.2**5
      RE(5)=IR(5)-IC(5)
      IO(5)=TEMPX(5)+ALPHA(5)*RE(5)
      IF(IABS(IO(5)).GT.8000)IO(5)=ISIGN(8000,IO(5))
      IF(ISD2A(IO(5),5).LT.0)ISTAT(3)=ISTAT(3).OR.2**5
C
      IF(IRTP(6,IC(6)).LT.0)ISTAT(2)=ISTAT(2).OR.2**6
      RE(6)=IR(6)-IC(6)
      IO(6)=TEMPX(6)+ALPHA(6)*RE(6)
      IF(IABS(IO(6)).GT.8000)IO(6)=ISIGN(8000,IO(6))
      IF(ISD2A(IO(6),6).LT.0)ISTAT(3)=ISTAT(3).OR.2**6
C
      IF(IRTP(7,IC(7)).LT.0)ISTAT(2)=ISTAT(2).OR.2**7
      RE(7)=IR(7)-IC(7)
      IO(7)=TEMPX(7)+ALPHA(7)*RE(7)
      IF(IABS(IO(7)).GT.8000)IO(7)=ISIGN(8000,IO(7))
      IF(ISD2A(IO(7),7).LT.0)ISTAT(3)=ISTAT(3).OR.2**7
C
      IF(IRTP(8,IC(8)).LT.0)ISTAT(2)=ISTAT(2).OR.2**8
      RE(8)=IR(8)-IC(8)
      IO(8)=TEMPX(8)+ALPHA(8)*RE(8)
      IF(IABS(IO(8)).GT.8000)IO(8)=ISIGN(8000,IO(8))
      IF(ISD2A(IO(8),8).LT.0)ISTAT(3)=ISTAT(3).OR.2**8
C
      IF(IRTP(9,IC(9)).LT.0)ISTAT(2)=ISTAT(2).OR.2**9
      RE(9)=IR(9)-IC(9)
      IO(9)=TEMPX(9)+ALPHA(9)*RE(9)
      IF(IABS(IO(9)).GT.8000)IO(9)=ISIGN(8000,IO(9))
      IF(ISD2A(IO(9),9).LT.0)ISTAT(3)=ISTAT(3).OR.2**9
C
      IF(IRTP(10,IC(10)).LT.0)ISTAT(2)=ISTAT(2).OR.2**10
      RE(10)=IR(10)-IC(10)
      IO(10)=TEMPX(10)+ALPHA(10)*RE(10)
      IF(IABS(IO(10)).GT.8000)IO(10)=ISIGN(8000,IO(10))
      IF(ISD2A(IO(10),10).LT.0)ISTAT(3)=ISTAT(3).OR.2**10
C
      IF(IRTP(11,IC(11)).LT.0)ISTAT(2)=ISTAT(2).OR.2**11
      RE(11)=IR(11)-IC(11)
      IO(11)=TEMPX(11)+ALPHA(11)*RE(11)
      IF(IABS(IO(11)).GT.8000)IO(11)=ISIGN(8000,IO(11))
      IF(ISD2A(IO(11),11).LT.0)ISTAT(3)=ISTAT(3).OR.2**11
C
      TEMPX(1)=TEMPX(1)+BETA(1)*RE(1)
      TEMPX(2)=TEMPX(2)+BETA(2)*RE(2)
```

```
      TEMPX(3)=TEMPX(3)+BETA(3)*RE(3)
      TEMPX(4)=TEMPX(4)+BETA(4)*RE(4)
      TEMPX(5)=TEMPX(5)+BETA(5)*RE(5)
      TEMPX(6)=TEMPX(6)+BETA(6)*RE(6)
      TEMPX(7)=TEMPX(7)+BETA(7)*RE(7)
      TEMPX(8)=TEMPX(8)+BETA(8)*RE(8)
      TEMPX(9)=TEMPX(9)+BETA(9)*RE(9)
      TEMPX(10)=TEMPX(10)+BETA(10)*RE(10)
      TEMPX(11)=TEMPX(11)+BETA(11)*RE(11)
C
C
C
C     INSERT DDC CODE BEFORE HERE
C
C
C     THE FOLLOWING CODE READS CHANNEL 12 FOR GENERAL USE
      IF(IRTP(12,IC(12)).LT.0)ISTAT(2)=ISTAT(2).OR.2**12
C
C     THE FOLLOWING CODE FORCES OPEN LOOP OTPUT
C     FOR CHANNEL 12 FOR THE AIR PRESSURE
      IF(ISD2A(IR(12),12).LT.0)ISTAT(3)=ISTAT(3).OR.2**12
      GOTO 10
9999  END


      FUNCTION IWAIT
      COMMON IN(2),IOUT(2)
      COMMON IR(12),IC(12),IO(12),RE(12)
      COMMON ALPHA(12),BETA(12),TEMPX(12)
      COMMON ISTAT(12),ITICKS
      DATA ICNT/0/
C
C     HANDLE THE DIGITAL IN AND OUT HERE
      IN(1)=IDIN(1)
      IN(2)=IDIN(2)
      CALL IDOUT(1,IOUT(1))
      CALL IDOUT(2,IOUT(2))

C
C     **NOTE LINK ERRS IGNORED FOR NOW**
C     ASSUME SUCCESSFUL
      IWAIT=0
C     LINK ROUTINES
      IF(ISND(IN))30,30,20
20    IF(ISND())30,30,20
C     NEXT 3 LINES HANDLES DDC/FUN GEN ITTERATION RATIO
30    ICNT=ICNT-1
      IF(ICNT.GT.0)GOTO 50
      ICNT=ISTAT(8)
C
      IF(IREC(IN))50,50,40
40    IF(IREC())50,50,40
50    CONTINUE
C
C     CLOCK WAIT ROUTINE AFTER HERE
C
6
```

```
 0      IS=ICLK0()
        IF(IS.EQ.0) GOTO 60
        IWAIT=IS-1
C       NEXT 3 LINES HANDLES DYNAMIC ALTERATION OF CLOCK RATE
        IF(ITICKS.EQ.ISTAT(7))RETURN
        ITICKS=ISTAT(7)
        CALL ICLK0(ITICKS)
        RETURN
C
        END
*
```

```
 TT:<S.PRO
C       S.FOR PROGRAM FOR AMAVS 1976 VERSION 3 8-DEC-76
        BLOCK DATA
        COMMON IN(2),IOUT(2)
        COMMON IR(12),IC(12),IO(12),RE(12)
        COMMON ALPHA(12),BETA(12),TEMPX(12)
        COMMON ISTAT(12),ITICKS
        DATA IN/2*0/,IOUT/2*0/
        DATA IR/12*0/,IC/12*0/,IO/12*0/,RE/12*0./
        DATA ALPHA/12*0./,BETA/12*0./,TEMPX/12*0./
        DATA ISTAT/6*0,100,5*0/,ITICKS/100/
        END


C       BEGIN MAINLINE OF DDC
        COMMON IN(2),IOUT(2)
        COMMON IR(12),IC(12),IO(12),RE(12)
        COMMON ALPHA(12),BETA(12),TEMPX(12)
        COMMON ISTAT(12),ITICKS
C
C       BEGIN EXECUTABLE CODE
C
C       START CLOCK
        CALL ICLK0(ITICKS)
C
C       START PRIMARY LOOP
10      CONTINUE
C
C       OPTIONAL DEBUG CODE FOR TIMING PURPOSES
C       USE SCOPE TO DETERMINE ITTERATION RATE & RATIO
C       OF WAIT TO COMPUTATION TIME
D               CALL ISD2A(0,13)
C
C       SYNC ON TIME BASE
        ISTAT(1)=IWAIT()
C
C       SECOND PART FOR TIMING
D               CALL ISD2A(4000,13)
C
C
C
C       INSERT DDC CODE AFTER HERE
C
C
C
C       INSERT DDC CODE BEFORE HERE
C
C
C       THE FOLLOWING CODE READS CHANNEL 12 FOR GENERAL USE
        IF(IRTP(12,IC(12)).LT.0)ISTAT(2)=ISTAT(2).OR.2**12
C
C       THE FOLLOWING CODE FORCES OPEN LOOP OTPUT
C       FOR CHANNEL 12 FOR THE AIR PRESSURE
        IF(ISD2A(IR(12),12).LT.0)ISTAT(3)=ISTAT(3).OR.2**12
        GOTO 10
9999    END
```

```
      FUNCTION IWAIT
      COMMON IN(2),IOUT(2)
      COMMON IR(12),IC(12),IO(12),RE(12)
      COMMON ALPHA(12),BETA(12),TEMPX(12)
      COMMON ISTAT(12),ITICKS
      DATA ICNT/0/
C
C     HANDLE THE DIGITAL IN AND OUT HERE
      IN(1)=IDIN(1)
      IN(2)=IDIN(2)
      CALL IDOUT(1,IOUT(1))
      CALL IDOUT(2,IOUT(2))

C
C     **NOTE LINK ERRS IGNORED FOR NOW**
C     ASSUME SUCCESSFUL
      IWAIT=0
C     LINK ROUTINES
      IF(ISND(IN))30,30,20
20    IF(ISND())30,30,20
C     NEXT 3 LINES HANDLES DDC/FUN GEN ITTERATION RATIO
30    ICNT=ICNT-1
      IF(ICNT.GT.0)GOTO 50
      ICNT=ISTAT(8)
C
      IF(IREC(IN))50,50,40
40    IF(IREC())50,50,40
50    CONTINUE
C
C     CLOCK WAIT ROUTINE AFTER HERE
C
60    IS=ICLK0()
      IF(IS.EQ.0) GOTO 60
      IWAIT=IS-1
C     NEXT 3 LINES HANDLES DYNAMIC ALTERATION OF CLOCK RATE
      IF(ITICKS.EQ.ISTAT(7))RETURN
      ITICKS=ISTAT(7)
      CALL ICLK0(ITICKS)
      RETURN
C
      END
*
```

```
 TT:<SGEN.BAS
100 PRINT 'NOTE THIS PROGRAM DESTROYS S.FOR'
110 PRINT '----PROCED WITH CAUTION'
120 PRINT 'NUMBER OF CHANNELS ';
130 INPUT N
205 OPEN 'S.FOR' FOR OUTPUT AS FILE #2
220 OPEN 'S.PRO' FOR INPUT AS FILE #1
230 IF END #1 THEN 900
240 INPUT #1:A$
250 PRINT #2:A$
260 IF A$<>'C   INSERT DDC CODE AFTER HERE'GO TO 230
320 FOR I=1 TO N
322 A$=''\B$=''\PRINT #2:'C'
325 I$=STR$(I)
327 A$='IF(IRTP('&I$&',IC('&I$&')).LT.0)ISTAT(2)=ISTAT(2).OR.2**'&I$
328 GOSUB 9000
330 A$='RE('&I$&')=IR('&I$&')-IC('&I$&')'
340 GOSUB 9000
350 A$='IO('&I$&')=TEMPX('&I$&')+ALPHA('&I$&')*RE('&I$&')'
360 GOSUB 9000
390 A$='IF(IABS(IO('&I$&')).GT.8000)'
395 B$='IO('&I$&')=ISIGN(8000,IO('&I$&'))'
400 GOSUB 9000
405 B$=''
410 A$='IF(ISD2A(IO('&I$&'),'&I$&').LT.0)ISTAT(3)=ISTAT(3).OR.2**'&I$
420 GOSUB 9000
490 NEXT I
499 A$=''\PRINT #2:'C'
500 FOR I=1 TO N
505 I$=STR$(I)
510 A$='TEMPX('&I$&')=TEMPX('&I$&')+BETA('&I$&')*RE('&I$&')'
520 GOSUB 9000
550 B$=''
590 NEXT I
600 IF END #1 THEN 990
610 INPUT #1:A$
620 PRINT #2:A$
630 GO TO 600
899 GO TO 990
900 PRINT 'ERROR - MISSING LINE IN PROTOTYPE FILE C  INSERT DDC ...
990 CLOSE
999 END
4490 NEXT I
9000 PRINT #2:'  '&A$&B$
9010 RETURN
*
```

```
 TT:<IGSWR.MAC
        .TITLE IGSWR
        .GLOBL IGSWR
        .MCALL .REGDEF
        .REGDEF
IGSWR:
        TST     (R5)+           ;IGNORE #PARAMS
        MOV     @#177570,R0     ;GET SWR
        RTS     PC              ;RETURN
        .END
*
```

```
 TT:<IDIN.MAC
        .TITLE DIG IN
;GLOBAL REFERENCES
        .GLOBL IDIN
;MACRO LIB CALLS
        .MCALL .REGDEF
        .REGDEF
;EXTERNAL PAGE DEFINITIONS
        CHAN0=171000
        CHAN1=171002
;IVAL=IDIN(ICHAN)
IDIN:
        CLR R0                    ;ASSUME ERROR

        CMP (R5)+,#1              ;ONE PARAM ONLY
        BNE RETURN

        MOV @(R5)+,R1    ;GET CHAN #
        BLE RETURN                ; ILLEGAL <=0(FOR CONS)

        CMP R1,#2                 ;LESS THAN MAX CHAN #?
        BGT RETURN                ; NO TOO BIG

        DEC R1                    ;FOR FOR SUBSCRIPT COMPAT

        ASL R1                    ;MAKE WORD INDEX

        MOV @TABLE(R1),R0         ;GET WORD

RETURN: RTS PC

;CONSTANT AREA
TABLE:  .WORD CHAN0
        .WORD CHAN1
        .END
*
```

```
 TT:<SL.MAC
        .TITLE  SL
        .CSECT  SL
;
        .LIST   MEB
        .MCALL  .REGDEF,..V2..
        .MCALL  .INTEN
;
        .REGDEF
        ..V2..
;
;DR11B ADDRESS CONSTANTS
        DRWC=172410                     ;WORD COUNT
        DRBA=172412                     ;BUFFER ADDRESS
        DRST=172414                     ;STATUS & CONTROL
        DRDB=172416                     ;DATA BUFFER
        DRVEC=124                       ;VECTOR ADDRESS
;
;MISC CONSTANTS
        MONLOW=54                       ;BASE OF MONITOR
        OFFSET=270                      ;OFFSET TO ENTRY OF Q MGR
        HDERR=1                         ;HARD ERROR BIT
        EOF=20000                       ;END OF FILE BIT
;
        PSW=177776                      ;PSW ADDRESS
        PR7=340                         ;PRI=7 PSW EQUIV
        PR5=240                         ;PRI=5 PSW EQUIV
;






;MACRO DEFINITIONS
;
        .MACRO  DEBUG
        .ENDM
;
;
        .MACRO  LA AD,R
        MOV     PC,R
        ADD     #AD-.,R
        .ENDM
;
```

```
;LOAD POINT
LOADPT: .WORD   DRVEC           ;VECTOR ADDRESS
        .WORD   DRINT-          ;OFFSET TO ISR
        .WORD   PR7             ;ENTER AT PRI 7
LQE:    .WORD   0               ;LAST Q ENTRY ADDR
CQE:    .WORD   0               ;CURRENT Q ENTRY ADDR
;
;
;
XFER:   DEBUG
        MOV     CQE,R4          ;GET ADDR OF Q ELEMENT
        ADD     #6,R4           ;       POINT TO WC
        ASL     (R4)            ;       AND MAKE BC=2*WC
        BCC     ERR             ;IF READ (POS) ERROR
        MOV     #DRDB,R3        ;PLACE ADDR OF DR DATA BUF IN R3
;
NEXT:   INC     (R4)            ;BC=BC+1
        BGT     DONE            ;       IF > 0 THEN DONE
;
        MOVB    @-2(R4),R5      ;GET NEXT BYTE
        INC     -2(R4)          ;       (UPDATE BUFFER POINTER)
        BIC     #177400,R5      ;       MAKE IT 8 BITS ONLY
        MOV     R5,(R3)         ;       AND PUT IN LINK
;
        CLR     -(R3)           ;NOW LETS
        MOV     #13,(R3)+       ;       SIGNAL OTHER PROCESSOR
;
        MOV     #200,R5 ;AND SET UP FOR
DELAY:  DEC     R5              ;       A DELAY (ASSUMING USING ABS LDR)
        BNE     DELAY
        BR      NEXT            ;WHEN TIRED OF WAITING GET ANOTHER CHAR
;
;
ERR:    DEBUG
        MOV     CQE,R4          ;GET CURR QUE ENTRY
        BIS     #HDERR,@-(R4)   ;SET ERROR FLAG IN CSW
;
DONE:   DEBUG
        CLR     @#DRST  ;CLEAN UP
        CLR     @#DRWC
        CLR     @#DRBA
        CLR     @#DRDB
        MOV     (SP),-(SP)      ;LETS FAKE AN INTERRUPT
        MOV     @#PSW,2(SP)
        BIS     #PR7,@#PSW
        .INTEN  5,PIC
RETURN: DEBUG
        LA      CQE,R4
        MOV     @#MONLOW,R5
        JMP     @OFFSET(R5)
;
;
;
        BR      RETURN          ;IF ABORTED GO TO RETURN
D
```

```
 RINT:  HALT                            ;SHOULD BE NO INTERRUPTS IN THIS ONE
        .WORD   0                       ;USED BY MONITOR
DRSIZE=.-LOADPT
        .END    XFER
*
```

```
TT:<ITTOUT.MAC
        .TITLE ITTOUT
 GLOBL REFERENCES
        .GLOBL ITTOUT
 MACRO LIB CALLS
        .MCALL .REGDEF,.TTYOUT
        .REGDEF
;RELOCATABLE SECTION
        .CSECT
;CALL ITTOUT(ICHR)
ITTOUT:
        MOV (R5)+,R1       ;GET NUM CHARS
1$:     DEC R1             ;REDUCE CHAR COUNT
        BLT 2$             ;IF DONE LEAVE
        MOV @(R5)+,R0      ;IF NOT GET CHAR
        .TTYOUT            ;AND OUTPUT IT
        BR 1$              ;THEN TRY FOR ANOTHER

2$:     CLR R0             ;JUST FOR NEATNESS
        RTS PC             ;RETURN TO CALLER
        .END
*
```

```
 TT:<IDOUT.MAC
        .TITLE DIG OUT
;GLOBAL REFERENCES
        .GLOBL IDOUT
;MACRO LIB CALLS
        .MCALL .REGDEF
        .REGDEF
;EXTERNAL PAGE DEFINITIONS
        CHAN0=171004
        CHAN1=171006
;ISTAT=IDOUT(ICHAN,IVAL)
IDOUT:
        CLR R0                      ;ASSUME ERROR

        CMP (R5)+,#2                ;SAFETY PERCAUTION
        BNE RETURN                  ; CANT EVEN MASTER SIMPLICITY

        MOV @(R5)+,R1               ;GET CHAN NO
        BLE RETURN                  ; TOO SMALL

        CMP R1,#2                   ;LESS THAN MAX CHAN NO
        BGT RETURN                  ; NO-TOO BAD

        DEC R1                      ;MAKE FOR SUB COMPAT

        ASL R1                      ;MAKE WORD INDEX

        MOV @(R5)+,@TABLE(R1)       ;OUTPUT VALUE

RETURN: RTS PC

;CONSTANT  AREA
TABLE:  .WORD.CHAN0
        .WORD CHAN1
        .END
*
```

```
 TT:<IRTP.MAC
        .TITLE RTP ROUTINES (IRTP)
;GLOBAL REFERENCES
        .GLOBL IRTP
;MACRO LIB CALLS
        .MCALL .REGDEF
        .REGDEF
;EXTERNAL PAGE AND VECTOR DEFS
        TMCSR=164000
        TMWC=164002
        TMMUX=164004
        TMADR=164006
        TMVEC=174
        TMPRI=340
;MISE CONSTANTS
        TMO=20.                         ;TIMEOUT LOOP COUNT
;RELOCATABLE SECTION
        .CSECT
;ISTAT=IRTP(ICHAN,IVAR)
;       ICHAN=          CHANNEL NUMBER (INTEGER)
;       IVAR=           VARIABLE TO RECIEVE A2D VAL(INTEGER)
;       ISTAT=-1        TIME OUT HAS OCCURRED
;       ISTAT=-2        ILLEGAL # OF PARAMS
;       ISTAT=-3        RTP HARDWARE ERROR
;       ISTAT>=0        OK & NUMBER OF TIMEOUT  COUNTS TO GO
IRTP:
        CMP (R5)+,#2                    ;NO PARAMS = 2
        BNE ERR2                        ; NO THEN ERROR

        MOV @(R5)+,R1                   ;GET CHANNEL NO
        DEC R1                          ; & MAKE FORTRAN COMPAT
        BIC #177740,R1                  ; MAKE IT LEGAL

        MOV #TMADR,R2                   ;GET LAST IO PAGE ADDR FOR RTP

        MOV (R5),(R2)                   ;LOAD ADDR
        MOV R1,-(R2)                    ;LOAD CHAN #
        MOV #-1,-(R2)                   ;LOAD WC  -  START CONVERSION

        MOV #TMO,R0                     ;SET UP TIME OUT
        MOV (R5),R1                     ;GET ADDRESS
        TST -(R2)                       ;POINT TO TMCSR

1$:     TSTB (R2)                       ;CONV DONE?
        BMI 2$                          ; YES - THEN GO ON

        SOB R0,1$                       ;NO THEN CHECK TIMEOUT

        DEC R0                          ;ERROR SO SET TO -1
        BR RETURN                       ; AND LET WORLD KNOW OF TIMEOUT

2$:     TST (R2)                        ;CHECK FOR RTP ERROR
```

```
        BMI ERR3                ; FOUND ONE GO REPORT

RETURN: RTS PC                  ;AND RETURN

ERR2:   MOV #-2,R0              ;LET WORLD KNOW OF BADNESS
        BR RETURN

ERR3:   MOV #-3,R0              ;REPORT HARD ERR
        BR RETURN
         END
*
```

```
 TT:<LINKSL.MAC
        .TITLE BUS LINK  FOR SLAVE
;GLOBL REFERENCES
        .GLOBL ISND,IREC
;MACRO LIB CALLS
        .MCALL .REGDEF
        .REGDEF
;EXTERNAL PAGE AND VECTOR DEFS
        DAWC=172410
        DAADR=172412
        DACSR=172414
        DABUF=172416
        DAVEC=124
        DAPRI=340
;MISC CONSTANTS
        ;NOTE REINTERP OF FLAGS
        XMIT=2              ;TRANSMIT REQUEST
        REC=4               ;RECEIVE REQUEST
;ABSOLUTE SECTION
        .ASECT
        .=DAVEC
        DAINT
        DAPRI
;RELOCATABLE SECTION
        .CSECT
;ISTAT=IREC(IARRAY,NWORDS)
IREC:
        MOV #REC,R1         ;PUT REC REQ IN R1
        BR COMMON           ;GOTO COMMON CODE

;ISTAT=ISND(IARRAY,NWORDS)
ISND:
        MOV #XMIT,R1        ;PUT XMIT REQ IN R1

COMMON:
;       0 PARAMS - RETURN STATUS ONLY
;       1 PARAMS - DO SEND OR REC WITH WC FROM MASTER
;       2 PARAMS - VERIFY WC & SEND OR REC

        TST (R5)+           ;CHECK NO PARAMS
        BLE STARET          ; RETURN STATUS IF 0 PARAMS

        CLR R0              ;FOR STATUS RETURN

        TST STATUS          ;SEE IF ALREADY RUNNING
        BGT ERR1            ; TELL CALLER TO TRY LATER

        MOVB @#DACSR+1,R2          ;GET FLAG BITS
        COM R2              ;SND=REC REC=SND
        BIC #177771,R2      ;CLEAR NOISE
        CMP R1,R2           ;DOES MASTER WANT TO DO LIKEWISE
        BNE ERR2            ; NO THEN TELL CALLER

        MOV (R5)+,R3        ;R3=ARRAY ADDR
        BIC #1,R3           ; AND MAKE SURE EVEN

        CMP -4(R5),#1       ;1 PARAM CALL?
```

```
        BNE 1$            ; NO THEN ASSUME 2

        MOV @#DABUF,R4    ; YEP- PICK UP WC FROM MASTER
        BPL ERR3          ; ERROR IF IT WAS POS
        BR 2$             ;GOTO IT

1$:     MOV @(R5)+,R4     ;R4=
        NEG R4            ;      -WC
        BPL ERR3          ;IF GROSS ERROR

        CMP R4,@#DABUF    ;DOES WC AGREE
        BNE ERR4          ; NO THEN DONT XFER

;THINGS LOOK GO AT THIS POINT
2$:     MOV R1,STATUS     ;STATUS=SEND OR REC

        MOV R4,@#DAWC     ;-WC
        MOV R3,@#DAADR    ;ARRAY ADDR
        MOV CSR(R1),@#DACSR       ;TAKE APPROP ACTION

STARET: MOV STATUS,R0     ;RETURN STATUS TO USER
RETURN: RTS PC            ;RETURN

;ERROR HANDLER
ERR4:   DEC R0
ERR3:   DEC R0
ERR2:   DEC R0
ERR1:   DEC R0
        BR RETURN

;INTERRUPT SERVICE ROUTINE
DAINT:
        TST @#DACSR
        BPL 1$            ; NO IT'S OK

        MOV #-6,STATUS    ; YES-STATUS =ERROR
        BR DARTI                  ;RETURN

1$:     CLR STATUS        ;SET STATUS=SUC

DARTI:  CLR @#DACSR       ;DISABLE INT & OLD INFO
        RTI               ;RETURN

;IMPURE
STATUS: .WORD 0           ;HANDLER STATUS

;CONSTANTS
CSR:    .WORD 0           ;SHOULD NEVER BE USED
        .WORD 503         ;CYC,IE,XMIT(NEW),GO
        .WORD 105         ;RECEIVE IE,REC,GO
        .END
*
```

```
 TT:<M10KHZ.MAC
        .TITLE M10KHZ
;MASTER 10 KHZ CLOCK WAIT ROUTINE
;       PASSES ERROR ON OVERRUN
;GLOBAL REFERENCES
        .GLOBL M10KHZ,JOBBLK
;MACRO LIBRARY CALLS
        .MCALL .REGDEF,..V2..,.INTEN,.SPND,.RSUM
        .MCALL .SYNCH
        .REGDEF
        ..V2..
;EXTERNAL PAGE AND VECTOR DEFS
        KWCSR=172540
        KWBUF=172542
        KWVEC=104
        KWPRI=340
;MISC DEFS
        KWRATE=113                      ;10 KHZ INT REPT GO
;RELOCATABLE SECTION
        .CSECT
;ISTAT=M10KHZ(ITICKS)
;       ISTAT>0 CLOCK OVERRUN
;       ISTAT=0 SUCCESS
;       ISTAT<0 ERROR WHICH SHOULD BE FATAL
M10KHZ:
        CMP (R5)+,#1                    ;PARAMS=1?
        BNE ERR1                        ; NO - THEN ERROR


        CMP @(R5)+,OLDTIC               ;NEW = OLD?
        BEQ 1$                          ; YEP - THEN SKIP EXCEPTION CODE

        MOV @-(R5),R1                   ; NOP - GET A COPY OF #TICKS
        MOV R1,OLDTIC                   ;OLD=NEW
        MOV R1,@#KWBUF                  ;SET UP NEW TIME

        TST ONETIM                      ;HAVE WE BEEN HERE BEFORE
        BNE 1$                          ; YEP - SKIP INITIALIZATION

        INC ONETIM                      ;AVOID COMING HERE AGAIN

        TST JOBBLK+2                    ;MAKE SURE INITRT WAS CALLED
        BEQ ERR3                        ; FATAL ERROR HOPE HS IS WATCHING

        MOV JOBBLK,SYNBLK+2             ;PLACE JOB NO IN SYNC BLOCK


        MOV #KWINT,@#KWVEC              ;INITIALIZE THE VECTOR
        MOV #KWPRI,@#KWVEC+2            ; AND PRIORITY

        MOV #KWRATE,@#KWCSR             ;START THE CLOCK

1$:     INC SUSCNT                      ;INCREMENT SUSPEND COUNTER

        .SPND                           ;SUSPEND
```

```
        MOV ERRCNT,R0               ;RETURN ERRCNT TO USER
        CLR ERRCNT                  ;READY ERRCNT FOR NEXT TIME

RETURN: RTS PC                      ;RETURN TO MAINLINE

;ERROR ROUTINES
ERR3:   MOV #-30.,R0                ;SHOULD BE FATAL
        BR RETURN
ERR2:   MOV #-20.,R0                ;SHOULD BE FATAL
        BR RETURN
ERR1:   MOV #-10.,R0                ;SHOULD BE FATAL
        BR RETURN


;INTERRUPT SERVICE ROUTINE
KWINT:
        TST SUSCNT                  ;SHOULD WE IGNORE THIS
        BGT 1$                      ; NO - IT IS GOOD
        INC ERRCNT                  ;REPORT AS OVERRUN
        RTI                         ; ---IGNORE THIS ONE---

1$:     CMP SUSCNT,#1               ;IS IT WHAT WE WANT?
        BEQ 2$                      ; YES - THEN DONT REPORT
        INC ERRCNT                  ; NO - LET MAINLINE KNOW

2$:     DEC SUSCNT                  ;SUSPENDS =SUSPENDS -1

         INTEN 7                    ;LET MONITR IN ON THIS
         SYNCH #SYNBLK              ;MAKE SURE TALKING TO TASK
        BR ERR2                     ;WOOPS AN ERROR

        .IRP X,<R0,R1,R2,R3,R4>     ;SAVE REGS
        MOV X,-(SP)
        .ENDM
        .RSUM                       ;RESUME THE MAINLINE
        .IRP X,<R4,R3,R2,R1,R0>     ;RESTORE REGS
        MOV (SP)+,X
         ENDM

        RTS PC                      ;RETURN TO MONITR

;IMPURE AREA
SYNBLK: .WORD 0,0,0,0,0,-1,0        ;SYNC BLOCK
ERRCNT: .WORD 0                     ;ERROR COUNTER
SUSCNT: .WORD 0                     ;SUSPENDS - RESUMES
OLDTIC: .WORD 0                     ;OLD NO OF TICKS
ONETIM: .WORD 0                     ;ONE TIME ONLY FLAG
         END
*
```

```
 TT:CLINKRT.MAC
        .TITLE BUS LINK  FOR MASTER
;GLOBL REFERENCES
        .GLOBL MSND,MREC,JOBBLK
;MACRO LIB CALLS
        .MCALL .REGDEF,..V2..,.SPND,.SYNCH,.RSUM,.INTEN
        .REGDEF
        ..V2..
;EXTERNAL PAGE AND VECTOR DEFS
        DAWC=172410
        DAADR=172412
        DACSR=172414
        DABUF=172416
        DAVEC=124
        DAPRI=340
;MISC CONSTANTS
        ;NOTE REINTERP OF FLAGS
        XMIT=2          ;TRANSMIT REQUEST
        REC=4           ;RECEIVE REQUEST
;RELOCATABLE SECTION
        .CSECT
;ISTAT=MREC(IARRAY,NWORDS)
MREC:
        MOV #REC,R1     ;PUT REC REQ IN R1
        BR COMMON       ;GOTO COMMON CODE

;ISTAT=MSND(IARRAY,NWORDS)
MSND:
        MOV #XMIT,R1    ;PUT XMIT REQ IN R1

COMMON:

        CMP (R5)+,#2    ;TWO PARAMETER CALL?
        BNE STARET      ; NO THEN JUST RETURN STAT

        CLR R0          ;FOR STATUS RETURN

        TST STATUS      ;SEE IF ALREADY RUNNING
        BGT ERR1        ; TELL CALLER TO TRY LATER

        BIT #XMIT+REC*400,@#DACSR       ;SLAVE READY
        BNE ERR2                        ; NO REPORT SAME
        MOV (R5)+,R3    ;R3=ARRAY ADDR
        BIC #1,R3       ; AND MAKE SURE EVEN

        MOV @(R5)+,R4   ;R4=
        NEG R4          ;    -WC
        BPL ERR3        ;IF GROSS ERROR


;THINGS LOOK GO AT THIS POINT
        TST ONETIM              ;HAVE WE BEEN HERE BEFORE?
        BNE 1$                  ;-YEP THEN SKIP INIT CODE
        INC ONETIM              ;-NO THEN SET FLG
```

```
        MOV JOBBLK,SYNBLK+2         ;SET UP JOB NUMBER

        MOV #DAINT,@#DAVEC          ;SETUP VEC (MAKE 1 TIME LATER)
        MOV #DAPRI,@#DAVEC+2        ; &PRIORITY
1$:     MOV R1,STATUS     ;STATUS=SEND OR REC

        MOV R4,@#DAWC     ;-WC
        MOV R4,@#DABUF    ;-WC FOR OTHER PROCESSOR
        MOV R3,@#DAADR    ;ARRAY ADDR
        MOV CSR(R1),@#DACSR         ;TAKE APPROP ACTION

         SPND                       ;WAIT FOR COMPLETION

STARET: MOV STATUS,R0     ;RETURN STATUS TO USER
RETURN: RTS PC            ;RETURN

;ERROR HANDLER
ERR4:   DEC R0
ERR3:   DEC R0
ERR2:   DEC R0
ERR1:   DEC R0
        BR RETURN

;INTERRUPT SERVICE ROUTINE
DAINT:
        .INTEN 7                    ;NOTIFY MONITR

        TST @#DACSR
        BPL 1$            ; NO IT'S OK

        MOV #-6,STATUS    ; YES-STATUS =ERROR
        BR DARTI                    ;RETURN

1$:     CLR STATUS        ;SET STATUS=SUC

DARTI:  CLR @#DACSR       ;DISABLE INT & OLD INFO
        .SYNCH #SYNBLK              ;GET TO RIGHT JOB
        BR SYNERR         ;SYNC ERROR
        .IRP X,<R5,R4,R3,R2,R1,R0>
        MOV X,-(SP)
        .ENDM
        .RSUM                       ;RESUME MAINLINE
        .IRP X,<R0,R1,R2,R3,R4,R5>
        MOV (SP)+,X
        .ENDM
        RTS PC            ;RETURN

SYNERR: MOV #-7,STATUS    ;SYNC ERROR
        HALT              ;!!!!!!!!!!!!!!!!VERY BAD PROBLEM

;IMPURE
STATUS: .WORD 0           ;HANDLER STATUS
ONETIM: .WORD 0           ;ONETIME FLAG
SYNBLK: .WORD 0,0,0,0,0,-1,0

;CONSTANTS
CSR:    .WORD 0           ;SHOULD NEVER BE USED
```

```
        .WORD XMIT+100+1;REINTERP XMIT CODE
        .WORD REC+100+1 ;REINTERP REC CODE
        .END
*
```

```
 TT:<ISD2A.MAC
        .TITLE ISD2A.MAC
;GLOBAL REFERENCES
        .GLOBL ISD2A
;MACRO CALLS
        .MCALL .REGDEF
        .REGDEF
;EXTERNAL PAGE ADDRESES
        CHAN0=171010
;MICS CONSTANTS
        OFFSET=20000

;ISTAT=ISD2A(IVAL,ICHN)
ISD2A:
        CLR R0             ;ASSUME OK

        CMP (R5)+,#2       ;2 PARAMS?
        BNE ERR2           ; NO THEN ERROR

        MOV @(R5)+,R1      ;GET VALUE TO OUTPUT
        ADD #OFFSET,R1     ;ADD Y INTERCEPT
        BGT 1$             ;IF > 0 THEN OK

        CLR R1             ; IF NOT THEN SATURATE LOW
        DEC R0             ; AND ACKNOWLEDGE ERROR
        BR 2$              ;SKIP HIGHLIM CHECK

1$:     CMP R1,#2*OFFSET-1        ;HILIM EXCEEDED?
        BLT 2$             ; NO- ITS OK

        MOV #2*OFFSET-1,R1        ;YES -SATURATE HIGH
        DEC R0             ;NOTE ERRROR

2$:     MOV @(R5)+,R2      ;GET CHAN #
        DEC R2             ; & MAKE FOR COMPAT

        BIT #177740,R2     ;CHAN <=31. & NOT NEG
        BNE ERR3           ; OUT OF RANGE ERROR

        ASL R2             ;MAKE WORD INDEX

        BIS BISTAB(R2),R1         ;SET CHANNEL BITS
        MOV R1,@UDCTAB(R2)        ;OUTPUT VALUE

RETURN: RTS  PC

ERR3:   DEC R0
ERR2:   DEC R0
ERR1:   DEC R0
        BR RETURN
UDCTAB:
        .REPT 8.
        .REPT 4
        .WORD CHAN0
        .ENDR
        CHAN0=CHAN0+2
```

```
        .ENDR

BISTAB:
        .REPT 8.
        .WORD 0,40000,100000,140000
        .ENDR

        .END
*
```

```
 TT:<ICLK0.MAC
        .TITLE ICLK0
;SLAVE 10KHZ READ THEN RESET CLOCK ROUTINE
;GLOBAL REFERENCES
        .GLOBL ICLK0
;MACRO LIBRARY CALLS
        .MCALL .REGDEF
        .REGDEF
;EXTERNAL PAGE AND VECTOR DEFS
        KWCSR=172540
        KWBUF=172542
        KWVEC=104
        KWPRI=340
;MISC DEFS
        KWRATE=113                      ;INT REPT 10 KHZ GO
;ABSOLUTE SECTION
        .ASECT
        .=KWVEC
        KWINT
        KWPRI
;RELOCATABLE SECTION
        .CSECT
;ISTAT=ICLK0([ITICKS])
;       ITICKS  =NO OF 10KHZ TICKS FOR CLOCK BASE
;       ISTAT   >=0     NO OF TICKS SINCE LAST CALL
ICLK0:
        CMP (R5)+,#1    ;ONE PARAM?
        BNE RETURN      ; NO THEN JUST RETURN STATUS

        MOV @(R5)+,@#KWBUF      ;SET UP NO OF COUNTS
        MOV #KWRATE,@#KWCSR     ; GO TO IT

RETURN: MOV TICKS,R0    ;PICK UP TICK COUNT
        SUB R0,TICKS    ;NORMALLY ZERO TICK COUNTER

        RTS PC          ;RETURN


KWINT:  TST @#KWCSR     ;ANY ERRORS?
        BPL 1$

        MOV #-32000.,TICKS      ;YES - BAD PROBLEMS

1$:     INC TICKS       ;ADD 1 TO TICK COUNT
        RTI             ;RETURN

;IMPURE AREA
TICKS:  .WORD 0
        .END
*
```

```
  TT:<MCLK0.MAC
        .TITLE MCLK0
;SLAVE 10KHZ READ THEN RESET CLOCK ROUTINE
;GLOBAL REFERENCES
        .GLOBL MCLK0,JOBBLK
;MACRO LIBRARY CALLS
        .MCALL .REGDEF
        .REGDEF
;EXTERNAL PAGE AND VECTOR DEFS
        KWCSR=172540
        KWBUF=172542
        KWVEC=104
        KWPRI=340
;MISC DEFS
        KWRATE=113                      ;INT REPT 10 KHZ GO
;RELOCATABLE SECTION
        .CSECT
;ISTAT=MCLK0([ITICKS])
;       ITICKS  =NO OF 10KHZ TICKS FOR CLOCK BASE
;       ISTAT   >=0       NO OF TICKS SINCE LAST CALL
MCLK0:
        CMP (R5)+,#1        ;ONE PARAM?
        BNE RETURN          ; NO THEN JUST RETURN STATUS

        MOV @(R5)+,@#KWBUF          ;SET UP NO OF COUNTS
        MOV #KWRATE,@#KWCSR         ; GO TO IT

        TST ONETIM          ;SEE IF WE HAVE BEEN HERE
        BNE RETURN          ; & IF SO DO NORM RETURN

        INC ONETIM          ;IF WE HAVENT MAKE SURE IT DOESN'T HAPPEN AGAIN
        MOV #KWINT,@#KWVEC          ;INITIALIZE THE
        MOV #KWPRI,@#KWVEC+2        ; VECTOR

        TST JOBBLK+2        ;MAKE SURE HE CALLED INITRT
        BNE RETURN          ; & IF HE DID RETURN
        HALT                ;BUT IF HE DIDN'T KILL HIM

RETURN: MOV TICKS,R0        ;PICK UP TICK COUNT
        SUB R0,TICKS        ;NORMALLY ZERO TICK COUNTER

        RTS PC              ;RETURN


KWINT:  TST @#KWCSR         ;ANY ERRORS?
        BPL 1$

        MOV #-32000.,TICKS          ;YES - BAD PROBLEMS

1$:     INC TICKS           ;ADD 1 TO TICK COUNT
        RTI                 ;RETURN

;IMPURE AREA
TICKS:  .WORD 0             ;TICK COUNTER
ONETIM: .WORD 0             ;ONETIME FLAG
        .END
*
```

```
 TT:<INITRL\L\T.MAC
        .TITLE INITRT
;INITIALIZATION ROUTINES FOR RT11 COMPATABILITY
;       SPECIFICALLY DOES .PROTECT , .GTJB , .DEVICE
;       FOR DALL1 AND KW11P LOCAL DRIVERS
;GLOBL REFERENCES
        .GLOBL INITRT,JOBBLK

;MACRO LIB CALLS
        .MCALL .REGDEF,..V2..
        .MCALL .PROTECT,.GTJB,.DEVICE
        .REGDEF
        ..V2..
;EXTERNAL PAGE & VECTOR DEFS
        KWCSR=172540
        KWBUF=172542
        KWVEC=104
        KWPRI=300
        DAWC=172410
        DAADR=172412
        DACSR=172414
        DABUF=172416
        DAVEC=124
        DAPRI=240
;RELOCATABLE SECTION
         CSECT

;ISTAT=INITRT()
;       ISTAT=0 SUCCESS
;       ISTAT<0 ERROR
INITRT:
        CLR ERRCNT                 ;CLEAR ERROR CNTR
        .PROTECT #AREA,#KWVEC      ;PROTECT KW11P
        BCS ERR1                   ; IF ERROR LEAVE
        .PROTECT #AREA,#DAVEC      ;PROTECT VECT DA11B
        BCS ERR2                   ; IF ERROR LEAVE
        .PROTECT #AREA,#400        ;PROTECT FOR INTER TASK COMMUNICATIONS
        BCS ERR3                   ; IF ERROR LEAVE

        .GTJB #AREA,#JOBBLK        ;GET JOB PARAMS

        .DEVICE #AREA,#DEVBLK      ;PREPARE ^C

RETURN: MOV     ERRCNT,R0          ;UPDATE ISTAT FOR RETURN
        RTS     PC
;ERROR ROUTINES
ERR3:   DEC ERRCNT
ERR2:   DEC ERRCNT
ERR1:   DEC ERRCNT
        BR RETURN         ;RETURN TO MAINLINE

;CONSTANTS
DEVBLK: KWCSR,0           ;ON ^C CLEAR KW11P CSR
```

```
        DACSR,0             ;ON ^C CLEAR DA11B CSR
        0                   ;END OF LIST
;IMPURE AREA
ERRCNT: 0
AREA:   .BLKW 2             ;RT11 DIRECTIVE BLOCK
JOBBLK: .BLKW 8.            ;AREA FOR JOB STATUS
        .END
*
```

```
 TT:<ID2A.MAC
        .TITLE ID2A
        .GLOBL  ID2A
        .MCALL   REGDEF
        .REGDEF
;ISTAT=ID2A(IARRAY)
;       WHERE IARRAY IS A 12 ELEMENT INTEGER ARRAY
;       ISTAT=0 IF SUCCESSFUL
;       ISTAT=-1 IF FAIL
ID2A:
        CMP     (R5)+,#1        ;ONE PARAM?
        BEQ     1$              ; YEP THEN GOOD
;
        MOV     #-1,R0          ;NO -SET ERROR FLAG
        BR      XIT             ;& GO RETURN
;
1$:     MOV     (R5)+,R0        ;GET ADDR OF ARRAY
        .IRP    X,<176760,176410,176430>
        .IRP    Y,<0,1,2,3>
        MOV     (R0)+,@#Y*2+X   ;OUTPUT CHANNEL
        .ENDM
        .ENDM
        CLR     R0              ;MARK AS SUCCESSFUL
XIT:    RTS     PC
        .END
*
```

```
 TT:<TT1.MAC
          .TITLE TERMINAL SWAPPER
          .MCALL .REGDEF,.PRINT,.EXIT
          .REGDEF
          MONLOW=54
          TTYVEC=60
          NEWTTY=300
          BAUD6=412
          BAUD12=1012
          BAUD24=2012
          OFFSET=304
          FILL=56
START:    MOV #TTYVEC,R1
          MOV #NEWTTY,R2
          MOV #4,R3
LOOP:     MOV (R1)+,(R2)+
          DEC R3
          BNE LOOP
          MOV #175610,R1
          MOV @#MONLOW,R2
          MOV #4,R3
          ADD #OFFSET,R2
          CLR @(R2)           ;NEW TO TURN OFF INT
LOOP2:    MOV R1,(R2)+
          TST (R1)+
          DEC R3
          BNE LOOP2
          MOV @#MONLOW,R2
          MOV #360,342(R2)
          MOV #BAUD24,@#FILL
          .PRINT #MSG
          .EXIT
MSG:      .ASCIZ <7><7>/*THIS TERMINAL ACTIVE*/
          .END START
*
```

```
 TT:<TT0.MAC
         .TITLE TERMINAL SWAPPER
         .MCALL .REGDEF,.PRINT,.EXIT
         .REGDEF
         MONLOW=54
         TTYVEC=300
         NEWTTY=60
         BAUD6=412
         BAUD12=1012
         BAUD24=2012
         OFFSET=304
         FILL=56
START:   MOV #TTYVEC,R1
         MOV #NEWTTY,R2
         MOV #4,R3
LOOP:    MOV (R1)+,(R2)+
         DEC R3
         BNE LOOP
         MOV #177560,R1
         MOV @#MONLOW,R2
         MOV #4,R3
         ADD #OFFSET,R2
         CLR @(R2)        ;NEW TO CLEAR INT IE
LOOP2:   MOV R1,(R2)+
         TST (R1)+
         DEC R3
         BNE LOOP2
         MOV @#MONLOW,R2
         MOV #360,342(R2)
         MOV #BAUD24,@#FILL
         .PRINT #MSG
         .EXIT
MSG:     .ASCIZ <7><7>/*THIS TERMINAL ACTIVE*/
         .END START
*
```

```
 TT:<FAD.BAT
$JOB
$RT11
.R FORTRAN
*FAD,FAD=FA/U/E/D
.R LINK
*FAD/R,FAD=FAD,BLOCKD,INITRT,LINKRT,M10KHZ,ID2A/F
$EOJ
*


 TT:<S.BAT
$JOB
$RT11
.R FORTRAN
*S,S=S/E
.R LINK
*S,S=S,IRTP,ISD2A,LINKSL,ICLK0/F/I/L/C
*IDIN,IDOUT,IGSWR
*$SIMRT
*$12K
*
.R PIP
*A.=S.LDA
*SL:=A.
$EOJ
*


 TT:<FA.BAT
$JOB
$RT11
.R FORTRAN
*FA,FA=FA/U/E
.R LINK
*FA/R,FA=FA,BLOCKD,INITRT,LINKRT,M10KHZ,ID2A/F
$EOJ
*


 TT:<BA.BAT
$JOB
$RT11
.R FORTRAN
*BA,BA=BA/E/U
.R LINK
*BA,BA=BA,ITTOUT/F
$EOJ
*
```

```
 TT:<BLOCKD.BAT
$JOB
$RT11
 R FORTRA
*BLOCKD,BLOCKD=BLOCKD/U/E
$EOJ
*


 TT:<MAC.BAT
$JOB
$RT11
 R MACRO
*INITRT,INITRT/L:MEB=INITRT
*LINKRT,LINKRT/L:MEB=LINKRT
*M10KHZ,M10KHZ/L:MEB=M10KHZ
*LINKSL,LINKSL/L:MEB=LINKSL
*IRTP,IRTP/L:MEB=IRTP
*ISD2A,ISD2A/L:MEB=ISD2A
*ICLK0,ICLK0/L:MEB=ICLK0
*IDOUT,IDOUT/L:MEB=IDOUT
*IGSWR,IGSWR/L:MEB=IGSWR
*IDIN,IDIN/L:MEB=IDIN
*ITTOUT,ITTOUT/L:MEB=ITTOUT
*ID2A,ID2A/L:MEB=ID2A
*TT0,TT0/L:MEB=TT0
*TT1,TT1/L:MEB=TT1
*SL,SL/L:MEB=SL
.R LINK
*TT0,TT0=TT0
*TT1,TT1=TT1
$EOJ
*


 TT:<SD.BAT
$JOB
$RT11
 R FORTRAN
*SD,SD=S/E/D
.R LINK
*SD,SD=SD,IRTP,ISD2A,LINKSL,ICLK0/F/I/L/C
*IDIN,IDOUT,IGSWR
*$SIMRT
*$12K
*
 R PIP
*A.=SD.LDA
*SL:=A.
$EOJ
*
```

APPENDIX B
HARDWARE

TABLE B-1

MASTER MINICOMPUTER CONFIGURATION

$

MASTER COMPUTER

1.	PDP-11/40-CE	#1057® INCLUDING:	
	MF11-L	CORE MEMORY	1
	LC11-A	CONTROLLER	1
	LA30-P	DECWRITER	1
	PC11	READER/PUNCH	1
2.	DL11-C	INTERFACE	1
3.	MM11-L	MEMORY EXPANSION	2
4.	AD01-MA	ANALOG/DIGITAL CONVERTER	1
5.	A124	MULTIPLEX MODULE	3
6.	AH04	SAMPLE AND HOLD OPTION	1
7.	AH05	BIPOLAR OPTION	1
8.	AA11-D	DIGITAL/ANALOG CONTROL	3
9.	BA614	12 BIT D/A BI-POLAR	12
10.	DD11-B	SYSTEM UNIT FOR DF	3
11.	KE11-E	EXP ONST:MUL,DIV,ASH,ASHC	1
12.	KE11-F	ORIENTED FLOATING POINT	1
13.	KW11-L	REAL-TIME CLOCK	1
14.	KW11-P	PROGRAMMABLE REAL TIME CLOCK	1
15.	BM792-YB	RX11 FLOPPY BOOTSTRAP LOADER	1
16.	RK11J-DE	DISK CONTROLLER	1

*

® TRADEMARK, REGISTERED: DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

TABLE B-2

SLAVE MINICOMPUTER CONFIGURATION

$

SLAVE COMPUTER

1.	PDP-11/40-CA	®#5195 INCLUDING:	
	MF11-L	CORE MEMORY	1
	DL11-A	INTERFACE	1
	LT33-DC	TELETYPE	1
2.	PC11	READER/PUNCH	1
3.	MM11-L	CORE MEMORY	1
4.	BA11-ES	EXTENSION MOUNTING BOX	1
5.	H720-E	PDP 11/20 POWER SUPPLY	1
6.	DA11-BD	UNIBUS LINK	1
7.	KE11-E	EXP ONST:MUL,DIV,ASH,ASHC	1
8.	KE11-F	ORIENTED FLOATING POINT	1
9.	KW11-P	PROGRAMMABLE REAL TIME CLOCK	1
10.	UDC11	DIGITAL IN/OUT MASTER FILE	1
11.	DD02	FILE UNITS	2
12.	BA233	SIGNAL CONDITIONING MODULE	8
13.	BA633	D/A MODULE	8
14.	H738-A	POWER PANEL	2
15.	BW400	ISOLATED POWER SIG. COND.	4
16.	BM803	LATCHING RELAY MODULE	2
17.	BW741	CONTACT SENSE	2
18.	BC40-C-04	MOUNTING PANEL	12

*

® TRADEMARK, REGISTERED: DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

PDP-11/LOW LEVEL SIGNAL ACQUISITION/ REAL-TIME PROCESSOR (RTP)

The RTP unit is an extremely flexible amplifier per channel computer controlled multiplexer A to D conversion system with adjustable gain and filtering. This RTP unit was designed and fabricated by personnel of the Data Acquisition Group of the Structures Test Branch. This RTP unit has a resolution of 1.25 microvolts per bit to 1 millivolt per bit in a voltage range of 2.25 microvolts to 8 volts with variable or fixed steps. The code used is 13 bits plus sign's 2 complement binary. It has individual gain settings from 1 to 1000 and separate channel calibration. The system total data sampling rate is about 35KC. The speed is limited by amplifier settling time and line propagation time. The RTP system output is displayed in local mode on a digital meter and in remote mode is sent to the DDC minicomputer in digital form.

This device is connected between the PDP-11 unibus and two data subsystems. Either subsystem can be read by executing the appropriate select code prior to the initiation of a read operation. Any block of valid sequential PDP-11 bus addresses can be used as a read-in area. Note, however, that the PDP-11 must be addressed at even boundaries since exchanges take place in 16-bit words rather than 8-bit bytes.

1. Time Out

Any attempt to access a non-existent bus address will cause a time-out error. The time-out feature provides protection against hanging the bus as the result of a program (or hardware) error. The existence of a time-out error is indicated by the presence of A "1" in bit 15 of Control and Status Register TMCSR (bus address 764000). A time out condition can be cleared by loading A "0" into bit 15 of TMCSR or by executing a "System Initialize."

2. Data Transfers

Data can be read from either Channel Addressable or Fixed Sequential Subsystems in 16-bit parallel words. The detailed read-in procedure is as follows:

2.1 Channel Addressable Subsystems

(1) Load into register TMMUX (bus address 764012) a word representing the initial data channel number for the block to be read (bits 00-06), the channel amplifier gain setting (bits 07, 08, 13), and the subsystem selection (bits 14, 15).

(2) Load into register TMADR (bus address 764006) the initial memory location of the area in which the data are to be stored. The number must be even.

(3) Load into register TMWC the two's complement equivalent of the number of words to be transferred. Loading this number starts the "Read Subsystem" operation.

(4) If it is desired to interrupt the main program when the read operation has terminated, load a "1" into bit 06 of register TMCSR. The vector address for the interrupt is 174-176 and the priority level is fixed at seven by the hardware. The interrupt subroutine should clear the enable by loading a "0" into bit 06 of TMCSR before returning to the main program.

(5) Step 4 may be omitted in favor of monitoring the ready flag (bit 07 of TMCSR). When the read operation has terminated, the ready flag will switch from "0" to "1".

2.2 Fixed Sequential Subsystems

(1) Load into register TMMUX a word indicating the subsystem selection (bits 14, 15). The remaining bits of this register are not applicable to fixed sequential subsystems.

(2) Execute steps 2 through 5 of paragraph 2.1.

3. Summary of Register Bit Assignments

Bit 06: Interrupt Enable - When set to "1" immediately after initiation of a "Read Subsystem" operation, executes an interrupt when operation terminates. Should be cleared by interrupt subroutine. Can also be cleared by System Initialize. Read/Write.

Bit 07: Block Ready Flag - Monitors repeatedly during a "Read Subsystem" operation. When operation has terminated, flag will switch from "0" to "1". Is cleared by initiation of next read operation. Is set by "System Initialize" and "Time-Out" error. Read only.

Bit 15: Time Out Flag - Set by program error or hardware failure (attempting to access non-existent memory). When set, clears aborted bus cycle and returns control of bus to processor. Can be cleared under program control or by "System Initialize." Read/partial write.

TMWC = 764002

To initiate "Read Subsystem" operation, load register with the two's complement equivalent of the number of words to be read. When register overflows, read operation is automatically terminated and ready flag of EMCSR is set. Read/Write.

TMADR = 764006

Load with initial address of PDP-11 read-in area prior to initiation of "Read Subsystem" operation. Throughout operation, contents of register indicate PDP-11 has address at which next incoming word will be stored. Read/write.

TMMUX = 764004

Bits 00-06:

Channel Counter - Load this position of register with number of first subsystem channel to be transferred to the PDP-11. Throughout "Read Subsystem" operation, contents indicate number of next subsystem channel to be transferred to PDP-11 storage. Not used with fixed sequential subsystems. Write only.

Bits 07-08:

Normally loaded with the low and middle order bits of the desired channel gain designator. If necessary, can be added to channel counter (bits 00-06) to increase range to 000-777. Write only.

Bit 13:

High order bit of channel gain designator. Used with bits 08, 07 to select one of eight possible channel gain values. Write only.

Bit 14, 15:

To select subsystem #1, load <u>01</u> in bits 14 and 15, respectively. To select subsystem #2, load <u>10</u> in bits 14 and 15, respectively. Write only.

TEST STAND

The DDC hardware and software for this phase were tested on a three-channel test stand designed to simulate a portion of the Advanced Metallic Air Vehicle wing carry through structure. Two hydraulic jacks are connected to a rigid beam. The third hydraulic jack is connected to a relatively flexible torque arm attached to the same beam.

Basic elements of the test stand are shown in Figure B-1. The 1000-pound load hydraulic jack is 1-1/2 in. bore with 24" stroke. The 300-pound load hydraulic jack has a 1" bore and 24" stroke. All servo-valves are rated at 5 GPM with a 1K drop across them at full flow. Hydraulic line pressure for all runs was maintained at 3000 psi.

There are significant differences between Phase I and Phase II loading systems. Phase I loads were in tension only. Phase II loads are connected to provide both tension and compression on the same hydraulic jack. Phase I loads were attached to a flexible load with a long time constant. Phase II loads are paralleled on a very stiff beam with a very short time constant (worst case).

LOAD-SENSING SYSTEM

Load cells for the two parallel connected hydraulic jacks are rated at a 1000-pound maximum. The load cell on the flexible arm is rated at 300 pounds maximum. Sensitivity of the load cells are 2 millivolt per volt for the 1000-pound load cells and 3 millivolts per volt for the 300-pound load cell. Full-scale signals (feedback) are 20 and 30 millivolts for the 1000- and 300-pound load cells respectively.

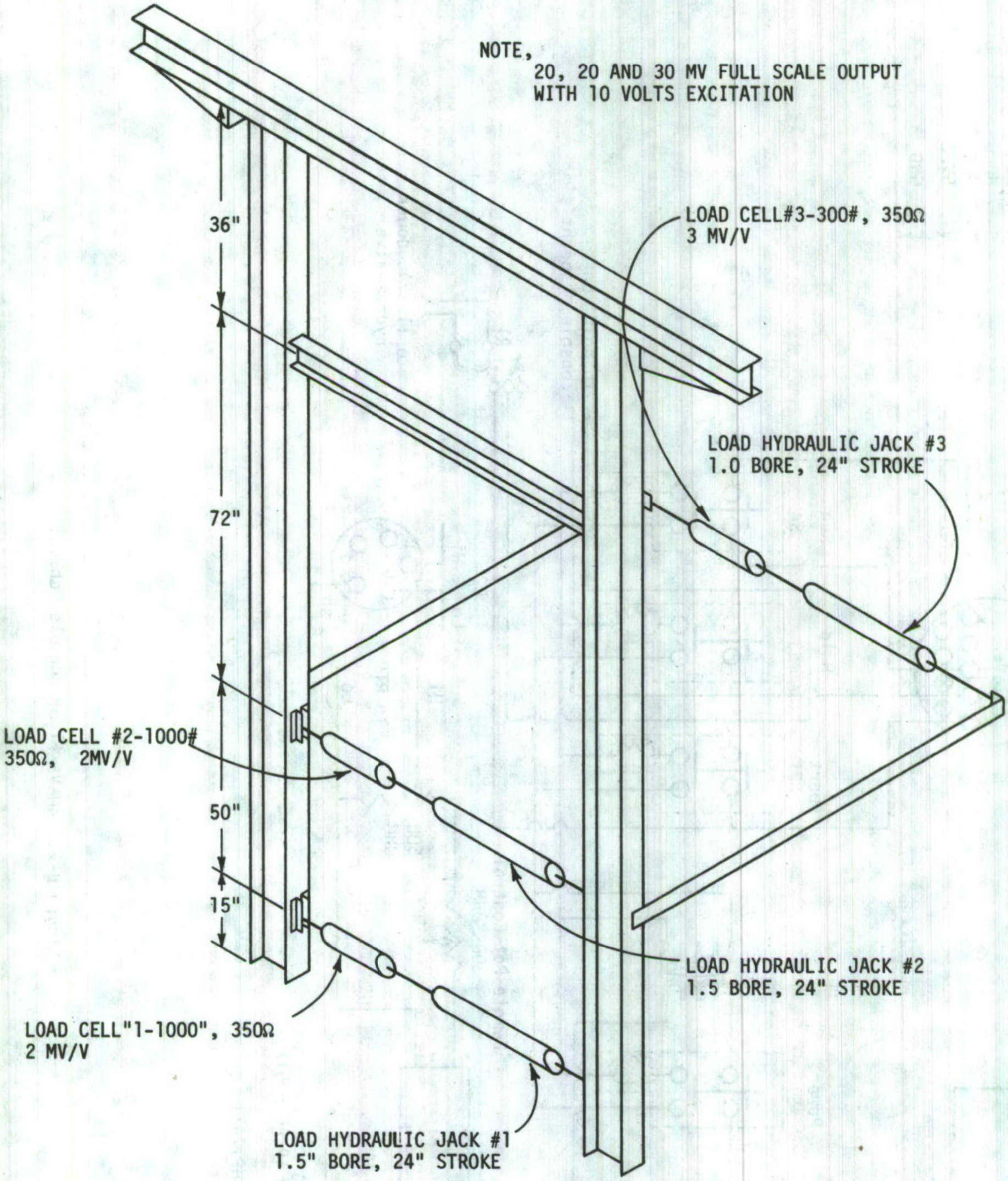


Figure B-1. Test Stand

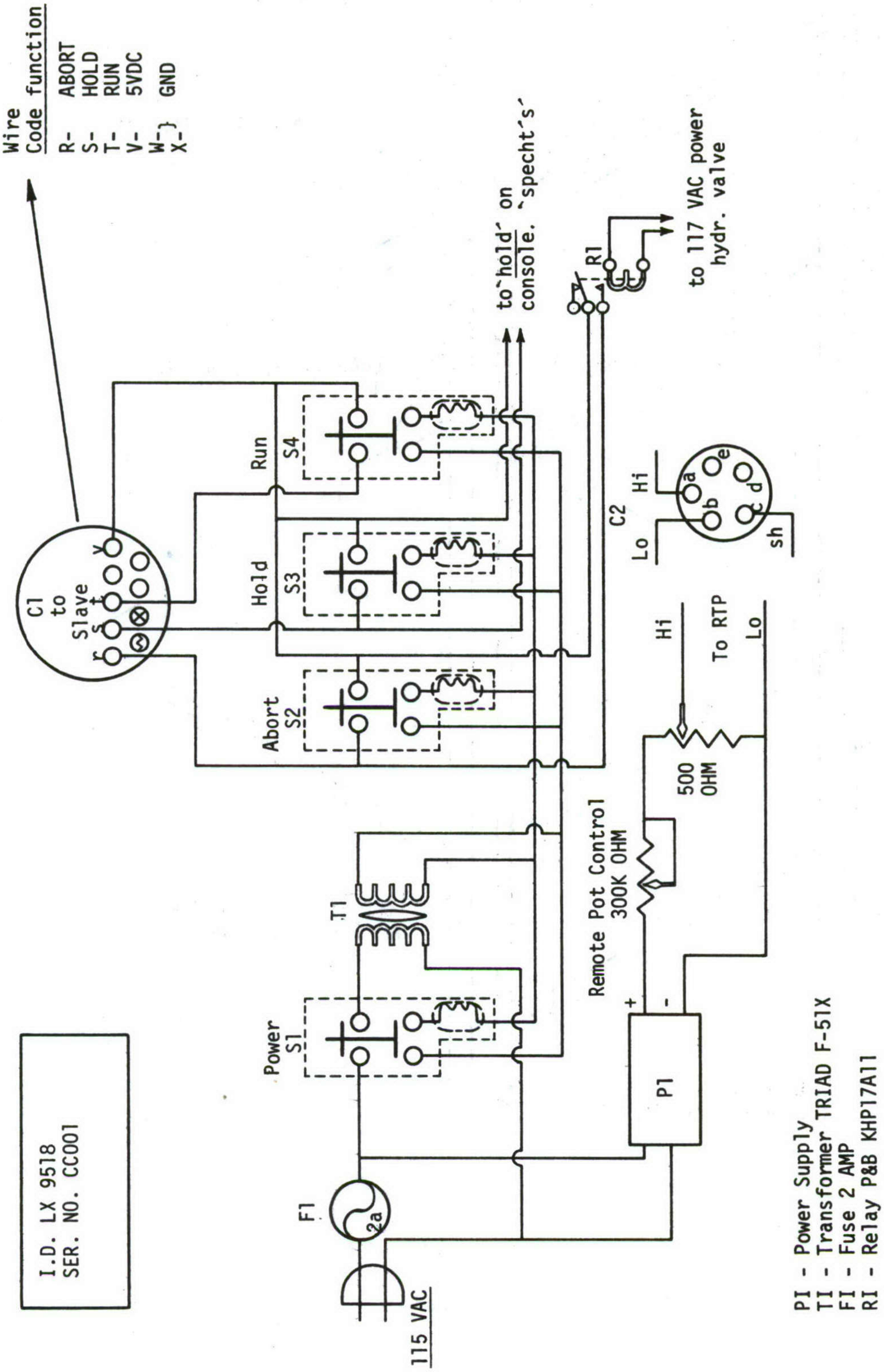


Figure B-2. AMAVS Floor Console Schematic

LOAD CELLS AND SIGNAL CONDITIONERS

LOAD CELLS

The load cells used to measure the applied load to the structure were four-bridge, universal type, strain gage based load cells of 350 ohm impedance manufactured by General Dynamics Corp., Forth Worth, Texas, for the Advanced Metallic Air Vehicle Structure test program. The load cells were calibrated to an accuracy of ±1% of full-scale output with a standard load cell traceable to NBS. The load cells used for DDC of the AMAVS had capacities of 100, 150, 200, and 575 KIPS with a nominal output of 5 millivolts per volt of input at full-scale load. A sketch of a typical load cell is shown in Figure B-3. The DDC system utilized the "D" bridge of each load cell.

SIGNAL CONDITIONING EQUIPMENT

Load cell signal conditioning modules were B&F Instruments, Inc. Model I-700 with 30-100K power supply and I-700SG Mode Card with ± one step 50K ohm shunt calibration. The outputs of the signal conditioning modules were fed into the Real Time Peripheral (RTP) unit.

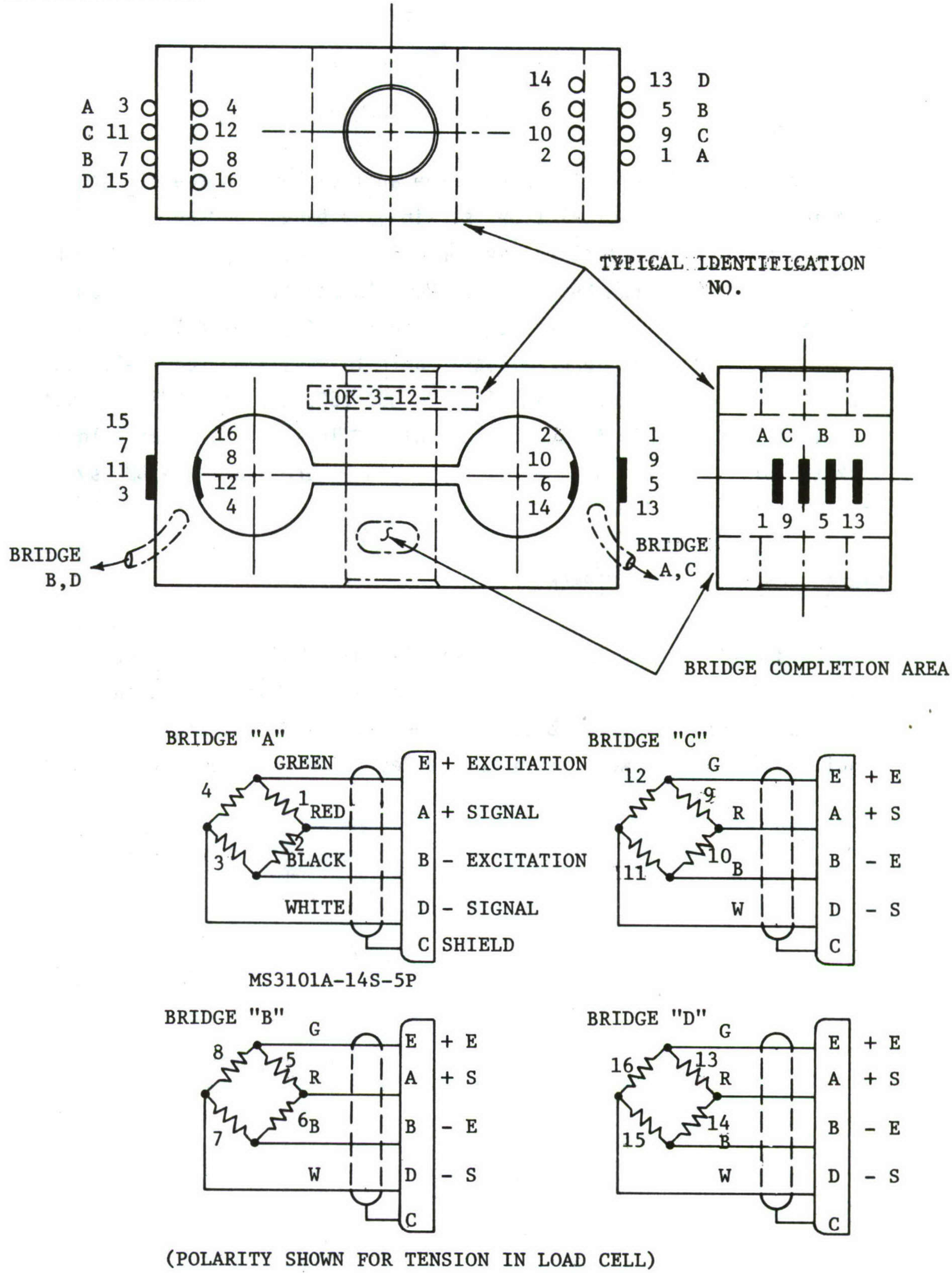
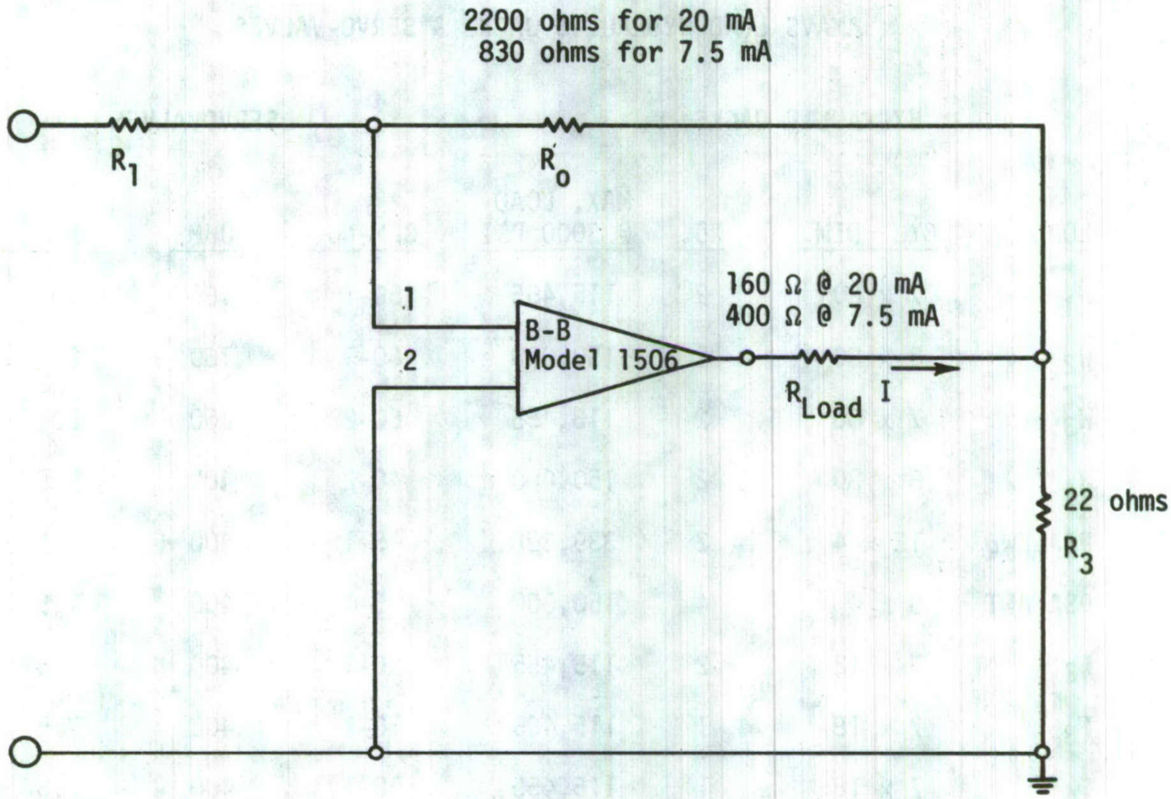
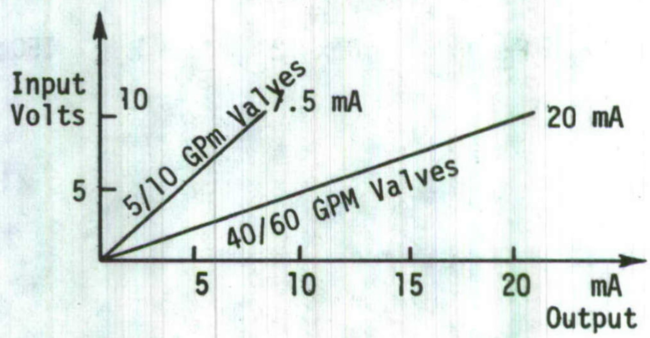


Figure B-3. AMAVS Load Cell Instrumentation Diagram



$$\frac{I}{E} = -\frac{R_o}{R_1 R_3} \quad \frac{\text{mA}}{\text{Volt}}$$



(Courtesy Burr-Brown, Operational Amp. Booklet)

Figure B-4. Servo-Valve Current Driver Schematic

TABLE B-3

AMAVS LOAD HYDRAULIC JACKS & SERVO-VALVES

	HYDRAULIC JACKS			SERVOVALVES		
LOAD	CYL. DIM.	NO.	MAX. LOAD @ 3000 PSI	GPM-NO.	OHMS	MA
W_1	7 x 80	2	115,455	60-1	160	20
W_2	8 x 80	2	150,000	60-1	160	20
W_3	7 x 80	2	115,455	60-2	160	20
W_4	8 x 80	2	150,000	40-2	400	7.5
PSA-FWD	12 x 4	2	339,300	5-1	400	7.5
PSA-AFT	8 x 4	4	150,000	5-1	400	7.5
F_2	7 x 12	2	115,455	10-1	400	7.5
F_3	7 x 18	1	115,455	10-1	400	7.5
F_4	7 x 18	1	115,455	10-1	400	7.5
F_5 L&R	8 x 46	2	150,000	40-1	400	7.5
F_6	8 x 12	1	150,000	10-1	400	7.5

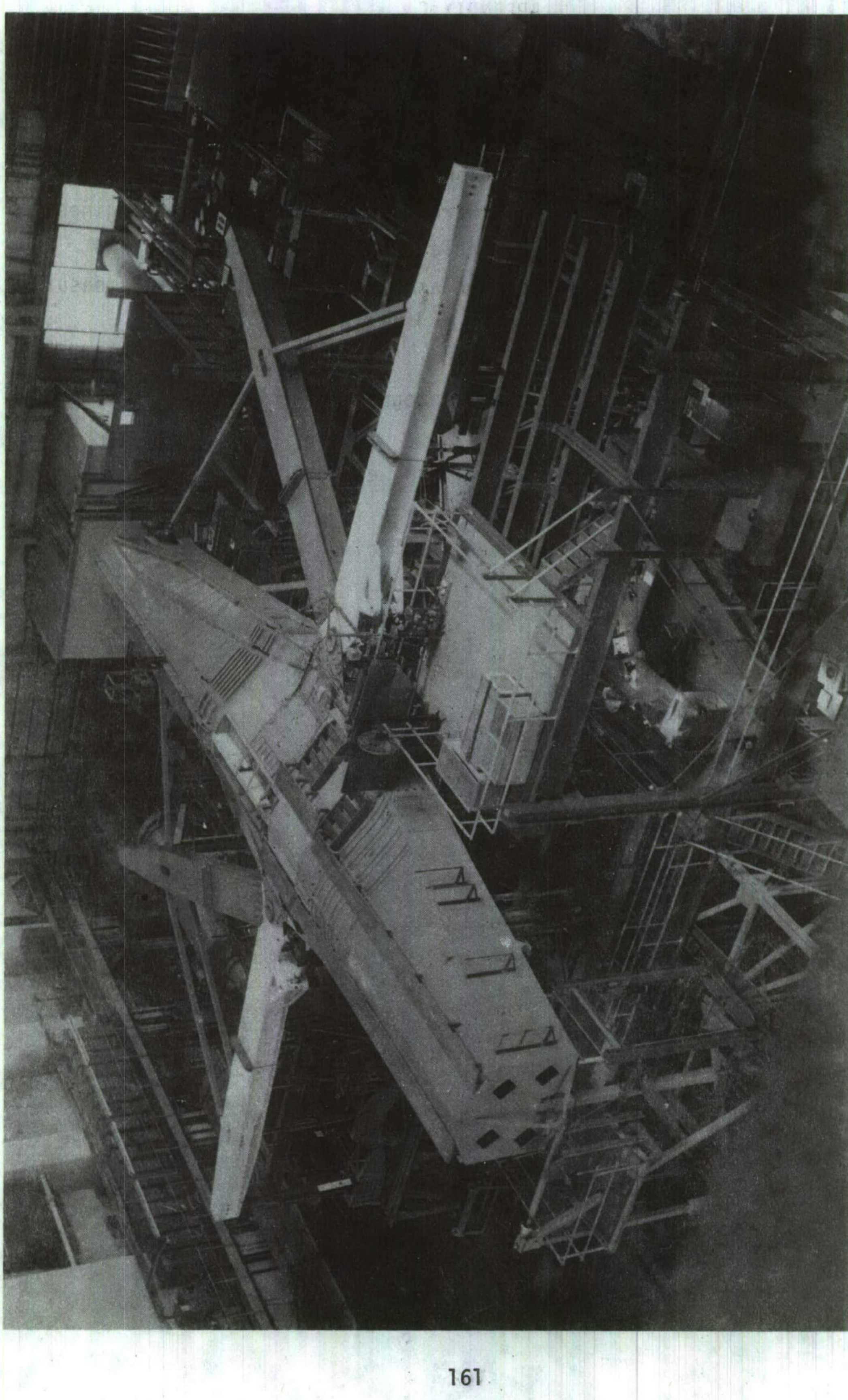


Figure B-5. View of AMAVS Test Article and Jig

APPENDIX C

DERIVATION OF DIGITAL CONTROLLER

The approach to controller design used here is to develop a digital algorithm that mimics analog proportional plus integral control action. This design procedure is straightforward and provides the test engineer with digital gains which are simply related to the analog gains and hence to analog observables (rise time and peak overshoot) which measure the effectiveness of the loop.

The continuous proportional plus integral controller is described by

$$M(t) = K_p e(t) + K_i \int_o^2 e(\tau)d\tau \qquad (A)$$

Let X(t) represent the integral, then, replacing the integration by the trapezoid rule yields

$$X(t_k) = X(t_{k-1}) + \frac{T}{2}(e(t_k) + e(t_{k-1})) \qquad (B)$$

Combining (B) with the sampled version of (A)

$$M(t_k) = K_p\ e(t_K) + K_i\ X(t_k)$$

provides a digital PI algorithm. This algorithm requires storage of the old value of the integral and the old value of the error. The storage of the error can be avoided by rearranging the computations as follows:

The difference equation corresponding to (B) and (A) is

$$M_k - M_{k-1} = (K_i T/2 + K_p)\ e_k + (K_i T/2 - K_p)\ e_{k-1}$$

The "observable" standard form simulation for this difference equation is [*]

$$X_{k+1} = X_k + K_i T\, e_k$$

$$M_k = X_k + (K_i T/2 + K_p)\, e_k$$

This is Darcy's algorithm with

$$B = K_i T$$

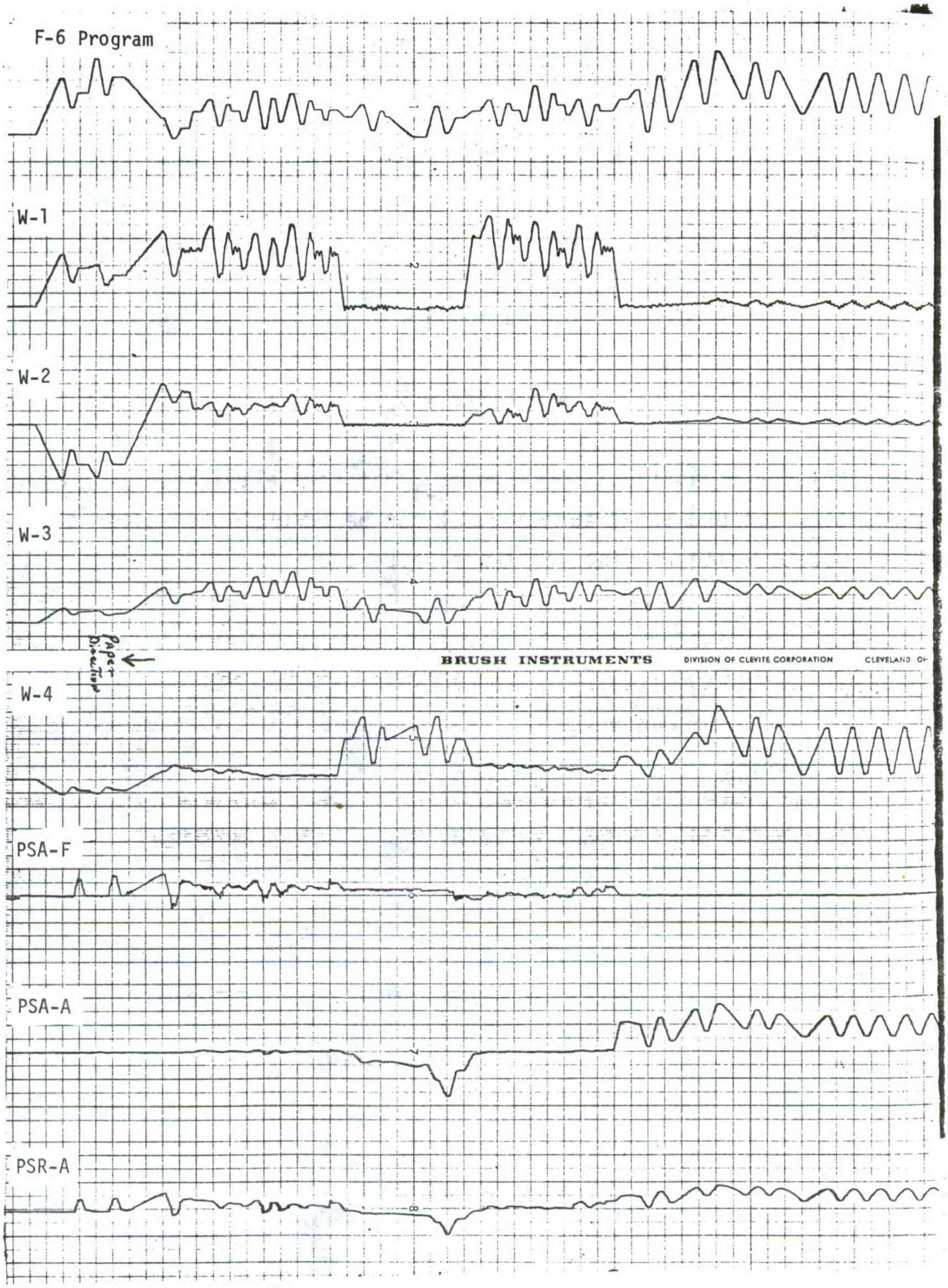
$$A = K_i T/2 + K_p$$

Notice that the requirement to store e_{k-1} has been eliminated and, more importantly, the update of the integral can be accomplished after the control update while waiting for the next period.

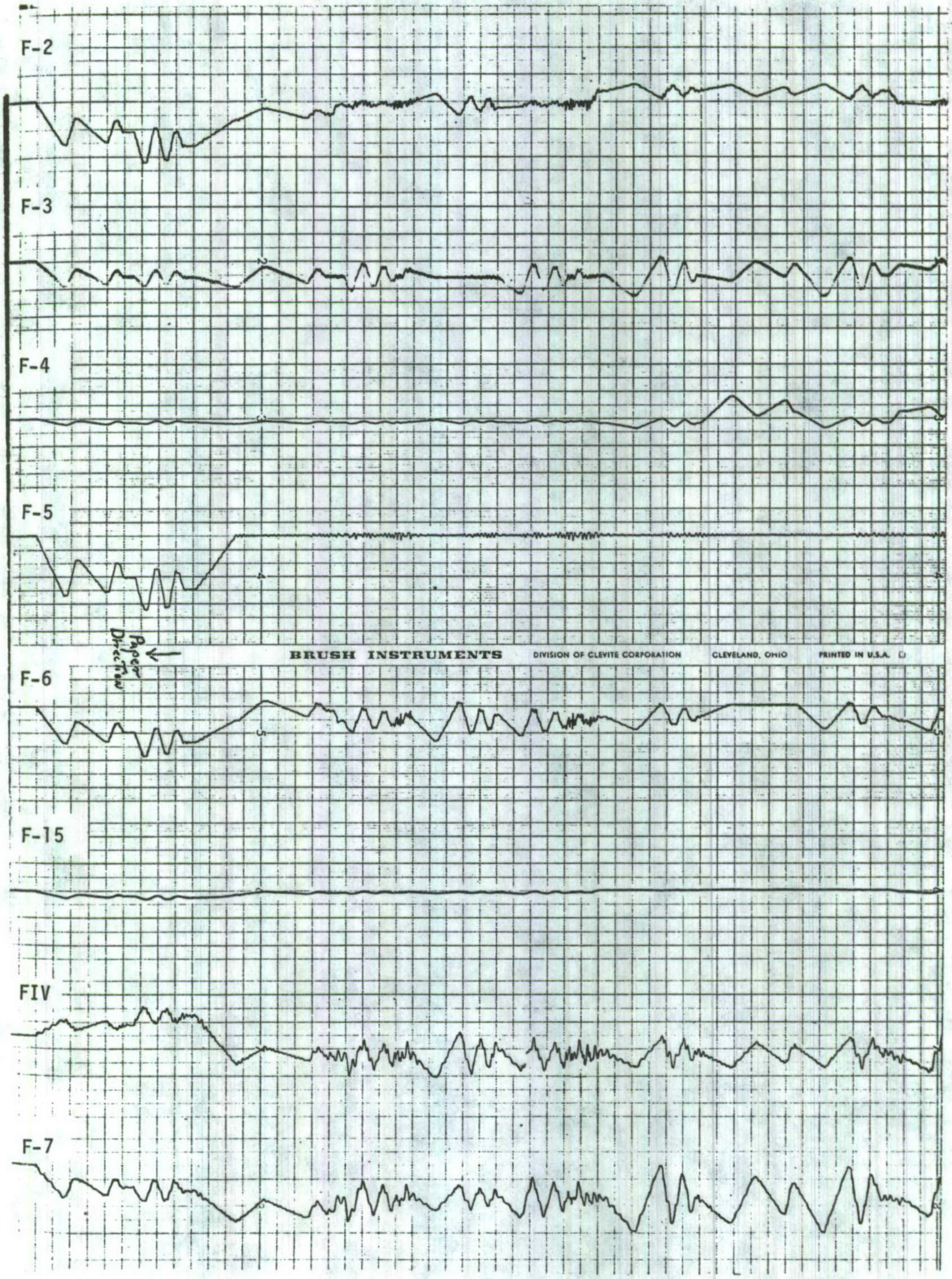
*State Variables for Engineers, De Russo, Roy, & Close, John Wiley & Sons, Inc., 1965.

APPENDIX D

LOAD PROFILE OF AMAVS FATIGUE TEST



APPENDIX D (Continued)



☆U.S.Government Printing Office: 1979 — 657-002/53